

# COOKIES(CIASCZKA)

## Czym jest cookie(ciaczka)?

Plik cookie to mały plik, który serwer umieszcza na komputerze użytkownika. Za każdym razem, gdy ten sam komputer żąda strony w przeglądarce, wysyła również plik cookie. Dzięki PHP możemy zarówno tworzyć, jak i pobierać wartości plików cookie. Plik cookie najczęściej jest używany do identyfikacji użytkownika.

## Tworzenie cookie(ciaczka) w PHP

W języku PHP żeby utworzyć cookie(ciaczka) trzeba użyć funkcji **setcookie()**

Składnia tej funkcji wygląda następująco: **setcookie(name, value, expire, path, domain, secure, httponly)**; Wymagany jest tylko parametr nazwy(name), a pozostałe są opcjonalne.

Ważna informacja: funkcja **setcookie()** musi się pojawić **przed** znacznikiem **<html>!**

### Przykład utworzenia cookie:

```
<?php
setcookie("user", "Admin", time() + 3600);
//Utworzenie cookie(ciaczka) o nazwie user oraz wartości Admin
//po 1 godzinie zostanie usunięte
?>
```

## Parametry funkcji setcookie() w PHP

**name** – nazwa cookie(ciaczka).

**value** – wartość którą cookie(ciaczka) będzie przechowywać.

**expire** – data kiedy cookie(ciaczka) wygaśnie.

**path** – ścieżka gdzie cookie(ciaczka) będzie dostępne.

**domain** – (Sub)domena, dla której cookie(ciaczka) jest dostępne.

**secure** – Wskazuje, czy cookie(ciaczka) powinno być przesyłane tylko przez bezpieczne połączenie HTTPS od klienta czy nie(true – przesyłane tylko przez HTTPS).

**httponly** – Kiedy true – cookie(ciaczka) będzie dostępne tylko przez protokół HTTP. Oznacza to, że cookie(ciaczka) nie będzie dostępne dla języków skryptowych, takich jak JavaScript.

## Pobranie wartości z cookie(ciaczka) w PHP

Pobrać wartość z cookie możemy za pomocą zmiennej globalnej **\$\_COOKIE**.

### Przykład pobrania wartości cookie:

```
<?php
setcookie("user", "Admin", time() + 3600);
//Utworzenie cookie(ciaczka) o nazwie user oraz wartości Admin
//po 1 godzinie zostanie usunięte
$wartosc = $_COOKIE["user"];
echo $wartosc;
?>
```

# Sprawdzenie czy cookie(ciasteczko) już istnieje

Aby dowiedzieć się, czy cookie(ciasteczko) jest już ustawiony używamy funkcji `isset()`.

*Przykład sprawdzenia czy cookie jest ustawione:*

```
<?php

setcookie("user", "Admin", time() + 20);
//Utworzenie cookie(ciasteczka) o nazwie user oraz wartości Admin
//po 20 sekundach zostanie usunięte
if(isset($_COOKIE["user"])){ //Sprawdzamy czy ciasteczko zostało ustawione
    $wartosc = $_COOKIE["user"];
    echo 'Wartość ciasteczka user = '.$wartosc;
}else{
    echo 'Ciasteczko nie jest ustawione';
}
?>
```

## Modyfikowanie cookie(ciasteczka) w PHP

Aby zmodyfikować cookie trzeba ponownie ustawić cookie za pomocą funkcji `setcookie()`.

*Przykład modyfikacji cookie:*

```
<?php

setcookie("user", "Admin", time() + 3600);
$wartosc = $_COOKIE["user"];
setcookie("user", "Adam", time() + 3600);
//Modyfikujemy cookie(ciasteczko) o nazwie user nadając mu wartość Adam. Te cookie po 1 godzinie
zostanie usunięte
echo 'Wartość ciasteczka user po zmianie = '.$wartosc;
?>
```

## Usuwanie cookie(ciasteczka) w PHP

Aby usunąć cookie trzeba użyć funkcji `setcookie()` z datą wygaśnięcia w przeszłości.

*Przykład usuwania cookie:*

```
<?php

setcookie("user", "Admin", time() + 3600);
setcookie("user", "Admin", time() - 3600); //Usuwanie cookie(ciasteczko)
if(isset($_COOKIE["user"])){
    $wartosc = $_COOKIE["user"];
    echo 'Wartość ciasteczka user = '.$wartosc;
}else{
    echo 'Ciasteczko nie istnieje(zostało usunięte)';
}
?>
```

# SESSIONS(SESJE)

## Czym jest sesja w PHP?

Sesja przechowuje w sobie **zmienne sesyjne** które natomiast **przechowują informacje** o **pojedynczym użytkowniku** i są **dostępne dla wszystkich stron** w jednej aplikacji.

## Rozpoczęcie sesji w PHP

Sesje **uruchamiamy** za pomocą funkcji `session_start()`. Teraz możemy **ustawić zmienne sesyjne** za pomocą globalnej zmiennej `$_SESSION`.

*Przykład uruchomienia sesji w pliku sesja1.php:*

```
<?php

session_start(); //Uruchomienie sesji
$_SESSION['imie'] = "Admin"; //Przypisanie do zmiennej sesyjnej imienia użytkownika
$_SESSION['ulubiony_kolor'] = "niebieski"; //Przypisanie do zmiennej sesyjnej ulubionego koloru
użytkownika
?>
```

## Uzyskanie wartości z sesji w PHP

Sesje **uruchamiamy** za pomocą funkcji `session_start()`. Teraz możemy **ustawić zmienne sesyjne** za pomocą globalnej zmiennej `$_SESSION`. **Żeby uzyskać** wartości zmiennych lokalnych **w innym pliku**, musimy w tym pliku **znowu otworzyć** sesję za pomocą funkcji `session_start()`.

*Przykład uzyskania wartości z sesji w pliku sesja2.php:*

```
<?php

session_start();
echo 'imie użytkownika: ' . $_SESSION['imie'] . ', a jego ulubiony kolor to: ' .
$_SESSION['ulubiony_kolor'];
?>
```

## Modyfikacja wartości z sesji w PHP

Żeby **zmodyfikować** dane wystarczy je **nadpisać**.

*Przykład modyfikacji wartości z sesji w pliku sesja2.php:*

```
<?php

session_start();
$_SESSION['imie'] = "Adam"; //Zmiana wartości zmiennej sesyjnej o nazwie imie z Admin na Adam
$_SESSION['ulubiony_kolor'] = "czerwony"; //Zmiana wartości zmiennej sesyjnej o nazwie
ulubiony_kolor z niebieskiego na czerwony
echo 'imie użytkownika: ' . $_SESSION['imie'] . ', a jego ulubiony kolor to: ' .
$_SESSION['ulubiony_kolor']; ?>
```

## Usunięcie sesji w PHP

Aby **usunąć wszystkie globalne zmienne sesji** i **zniszczyć sesję**, używamy funkcji `session_unset()` i `session_destroy()`.

*Przykład usunięcia sesji w pliku sesja1.php:*

```
<?php
```

```
session_start();
```

```
$_SESSION['imie'] = "Admin";
```

```
$_SESSION['ulubiony_kolor'] = "niebieski";
```

```
session_unset(); //Usuwa wszystkie zmienne sesyjne
```

```
session_destroy(); //Usuwa sesje
```

```
echo 'imie użytkownika: ' . $_SESSION['imie'] . ', a jego ulubiony kolor to:
```

```
' . $_SESSION['ulubiony_kolor']; //Tutaj wystąpi błąd wyświetlenia
```

```
?>
```