

Ciasteczka w PHP

Jak już wiemy z lekcji: [Obsługa ciasteczek w JS](#) ciasteczka to małe fragmenty danych, które przechowujemy w przeglądarce użytkownika, aby zapamiętać pewne informacje między wizytami na stronie. Dawniej były to małe pliki tekstowe, zaś we współczesnych przeglądarkach jest to lokalna baza danych SQLite na dysku klienta.

Po przepracowaniu obsługi ciasteczek w JavaScript, nauczymy się teraz tworzyć, odczytywać i usuwać ciasteczka w PHP. Zróbmy to prosto, jasno i z konkretnymi przykładami!

Jak tworzymy ciasteczko w PHP?

Tworzenie ciasteczka w PHP jest bardzo proste – używamy “gotowej” funkcji

`setcookie()` – jej składnia prezentuje się następująco:

```
setcookie(nazwa, wartość, czas_wygaśnięcia, ścieżka, domena, secure, httponly);
```

Najczęściej wystarczą trzy pierwsze argumenty, natomiast oczywiście omówmy teraz znaczenie wszystkich parametrów:

- **nazwa** – to klucz (nazwa ciasteczka), pod jakim będzie ono przechowywane. Musi być ciągiem znaków, najlepiej bez spacji i znaków specjalnych.
- **wartość** – wartość, jaką chcemy przypisać ciasteczku. Możemy tu np. przechować login użytkownika lub identyfikator sesji.
- **czas_wygaśnięcia** – określa, kiedy ciasteczko ma wygasnąć. Podajemy go w postaci znacznika czasu UNIX (np. `time() + 3600` ustawi ciasteczko na 1 godzinę od teraz). Jeśli pominiemy ten parametr, ciasteczko zostanie usunięte po zamknięciu przeglądarki.
- **ścieżka** – wskazuje, dla jakiej ścieżki na serwerze ciasteczko jest dostępne. Najczęściej ustawiamy `"/"`, aby ciasteczko było widoczne dla całej domeny.
- **domena** – domena, dla której ciasteczko ma obowiązywać. Możemy np. ograniczyć je tylko do subdomeny `"admin.mojserwis.pl"`.
- **secure** – parametr logiczny (true/false). Jeśli ustawimy na `true`, ciasteczko będzie przesyłane tylko przez połączenie HTTPS.
- **httponly** – również parametr logiczny. Jeśli ustawimy na `true`, ciasteczko nie będzie dostępne z poziomu JavaScript, co podnosi bezpieczeństwo (np. ochrona przed atakami XSS).

Przykład 1 – ustawiamy ciasteczko z imieniem użytkownika:

```
setcookie("imie", "Jolka", time() + 3600, "/");  
// Ciasteczko ważne przez 1 godzinę
```

Przykład 2 – stwórz ciasteczko z loginem użytkownika na 24 godziny:

```
setcookie("user", "Anna", time() + 86400, "/");  
// tworzy ciasteczko 'user' z wartością 'Anna', ważne przez 1 dobę
```

Co ważne – pamiętajmy: funkcję

`setcookie()` musimy wywołać **przed jakimkolwiek** wyjściem danych do przeglądarki (czyli zanim cokolwiek zostanie wypisane przez `echo`). W przeciwnym razie ciasteczko nie zostanie ustawione!

Obliczenia czasu w powyższych przykładach oczywiście bazują na dodaniu sekund do bieżącego czasu – 3600 sekund to godzina, zaś 86400 sekund odpowiada 24 godzinom – logiczne.

Jak odczytujemy ciasteczko w PHP?

Aby odczytać wartość zapisanej zmiennej, korzystamy z globalnej tablicy

`$_COOKIE`. Zrealizujmy zatem odczyt naszych ustawionych wcześniej ciasteczek!

Przykład 1 – wyświetlamy zapisane wcześniej imię:

```
if (isset($_COOKIE["imie"])) {  
    echo "Witaj ponownie, " . $_COOKIE["imie"] . "!";  
} else {  
    echo "Nie znaleziono ciasteczka."  
}
```

Przykład 2 – odczytanie wcześniej ciasteczka z loginem:

```
if (isset($_COOKIE["user"])) {  
    echo "Witaj ponownie, " . $_COOKIE["user"];  
} else {  
    echo "Ciasteczko 'user' nie zostało jeszcze ustawione."  
}
```

Zastosowaliśmy tutaj funkcję

`isset()`, aby sprawdzić, czy dane ciasteczko istnieje – w przeciwnym razie próba jego odczytu mogłaby wywołać błąd lub ostrzeżenie.

Warto wiedzieć, że tablica

`$_COOKIE` zawiera tylko te ciasteczka, które przeglądarka przesłała w żądaniu do serwera. Oznacza to, że ciasteczko ustawione przez `setcookie()` będzie dostępne dopiero **przy kolejnym odświeżeniu strony** lub następnym żądaniu HTTP. Pamiętajmy o tym na egzaminie i w codziennej pracy.

Jak usunąć ciasteczko w PHP?

Kolejna klasyczna mechanika – aby usunąć ciasteczko, wystarczy ustawić jego datę ważności w przeszłości – przeglądarka zauważy ten fakt i zwolni miejsce w bazie SQLite wyrzucając do kosza zgniłe ciasteczko 😊

Przykład 1 – usuwamy ciasteczko z imieniem:

```
setcookie("imie", "", time() - 3600, "/");  
// Ustawiamy datę wygaśnięcia na godzinę wstecz
```

Przykład 2 – usuwamy ciasteczko z loginem:

```
setcookie("user", "", time() - 3600);  
// Przy kolejnym odświeżeniu przeglądarka usunie ciasteczko 'user'
```

Wyzerowaliśmy wartość ciasteczka i ustawiliśmy czas jego wygaśnięcia na jedną godzinę wstecz. Przeglądarka usuwa takie ciasteczka automatycznie. Warto pamiętać, że – tak jak przy ustawianiu – usunięcie ciasteczka będzie widoczne dopiero przy następnym żądaniu HTTP.

Obsługa ciasteczek na jednej stronie

Pora na przykład “zbiorczy” – całość obsługi ciasteczek została poniżej zrealizowana w ramach jednego pliku PHP – w ten sposób najlepiej zrozumiemy umiejscowienie poszczególnych zapisów oraz dokonamy analizy całego dokumentu:

```
<?php
if (isset($_POST["zapisz"])) {
    setcookie("imie", $_POST["imie"], time() + 3600, "/");
    header("Location: ".$_SERVER["PHP_SELF"]);
    exit();
}
if (isset($_POST["usun"])) {
    setcookie("imie", "", time() - 3600, "/");
    header("Location: ".$_SERVER["PHP_SELF"]);
    exit();
}
?>
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <title>Ciasteczka w PHP</title>
</head>
<body>
    <h2>Ustaw imię w ciasteczku</h2>
    <form method="post">
        <input type="text" name="imie" required>
        <button type="submit" name="zapisz">Zapisz</button>
    </form>
    <h2>Odczytaj zapisane imię</h2>
    <p>
        <?php
        if (isset($_COOKIE["imie"])) {
            echo "Witaj ponownie, " . $_COOKIE["imie"] . "!";
        } else {
            echo "Brak zapisanego imienia.";
        }
        ?>
    </p>
    <h2>Usuń ciasteczko</h2>
    <form method="post">
        <button type="submit" name="usun">Usuń ciasteczko</button>
    </form>
</body>
</html>
```

W kodzie tej podstrony zaimplementowano trzy główne funkcjonalności:

- Ustawienie imienia w ciasteczku (po przesłaniu formularza),

- Odczytanie ciasteczka, jeśli już istnieje,
- Usunięcie ciasteczka – również za pomocą formularza.

Ustawianie ciasteczka:

```
if (isset($_POST["zapisz"])) {  
    setcookie("imie", $_POST["imie"], time() + 3600, "/");  
    header("Location: ".$_SERVER["PHP_SELF"]);  
    exit();  
}
```

Jeśli użytkownik kliknie przycisk **“Zapisz”**, formularz przesyła dane metodą POST. Warunek

`isset($_POST["zapisz"])` sprawdzi, czy formularz został przesłany. Jeśli tak, zapisujemy imię z pola tekstowego w ciasteczku o nazwie `"imie"`. Ponadto:

- Czas wygaśnięcia to `time() + 3600`, czyli dokładnie jedna godzina od “teraz”.
- `"/"` jako ścieżka oznacza, że ciasteczko jest dostępne w całym serwisie.
- Po ustawieniu ciasteczka wykonujemy **przekierowanie do tej samej strony**, bo efekt jest widoczny dopiero przy kolejnym żądaniu!

Usuwanie ciasteczka:

```
if (isset($_POST["usun"])) {  
    setcookie("imie", "", time() - 3600, "/");  
    header("Location: ".$_SERVER["PHP_SELF"]);  
    exit();  
}
```

Jeśli klikniemy przycisk **“Usuń ciasteczko”**, warunek się spełni i zostanie wywołana funkcja

`setcookie()`, ale z czasem wygaśnięcia ustawionym na godzinę wstecz. Efekt? Przeglądarka usunie takie ciasteczko automatycznie. Po operacji następuje ponowne załadowanie strony – myślę, że na tym etapie lekcji doskonale już wiemy dlaczego.

Wyświetlenie komunikatu na stronie:

```
if (isset($_COOKIE["imie"])) {  
    echo "Witaj ponownie, " . $_COOKIE["imie"] . "!";  
} else {  
    echo "Brak zapisanego imienia.";  
}
```

Na stronie znajduje się sekcja HTML z komunikatem. Jeśli ciasteczko istnieje, użytkownik widzi spersonalizowane powitanie. Jeśli nie – pojawia się informacja o jego braku.

To przykład bardzo dobrej praktyki – logika działania znajduje się u góry pliku (PHP), a część wizualna (formularze, teksty) w bloku HTML. Takie podejście sprawia, że kod jest przejrzysty, łatwy w utrzymaniu i wygodny do testowania.

Podsumowanie – co zapamiętać?

- W PHP ciasteczka tworzymy funkcją `setcookie()`, którą należy wywołać przed wysłaniem jakichkolwiek danych do przeglądarki.
- Kluczowe parametry funkcji `setcookie()` to: nazwa, wartość, czas wygaśnięcia (UNIX timestamp), ścieżka, domena, secure, httponly.
- Wartości ciasteczek odczytujemy z tablicy superglobalnej `$_COOKIE`; przed odczytem warto sprawdzić ich istnienie przez `isset()`.
- Aby usunąć ciasteczko, wywołujemy `setcookie()` z czasem wygaśnięcia ustawionym w przeszłości.
- Ciasteczko jest dostępne dopiero przy kolejnym żądaniu HTTP – dlatego po jego utworzeniu warto wykonać `header("Location: ...")` i przeładować stronę.
- Warto dbać o przejrzystość kodu: logika w PHP u góry pliku, interfejs (HTML) poniżej – pozwala to skutecznie testować działanie ciasteczek.