

Ciasteczka (cookies)

Ciasteczka (cookies) to niewielki fragment informacji, który aplikacja internetowa może zapisać na komputerze użytkownika.

Cookies (Ciasteczka) to małe pliki tekstowe przechowywane po stronie użytkownika, tworzone przez stronę internetową, które przechowują informacje.

Gdy przeglądarka łączy się ze stroną, w pierwszej kolejności szuka lokalnych cookies, odpowiednich dla danej witryny. Jeżeli znajdzie – zostaną one przekazane serwerowi.

Zastosowanie:

Użytkownik wchodzi na stronę, wyświetla się reklama, po odświeżeniu strony (ponownym wejściu na stronę) wyświetli się kolejna (inna) reklama, czyli użytkownik jest identyfikowany za pomocą ciasteczek.

Ciasteczka mają datę ważności, czyli moment, do którego są przechowywane na dysku twardym użytkownika.

W PHP możemy ustawić cookie za pomocą funkcji `setcookie()`.

Funkcja `setcookie()`

```
setcookie('nazwa', 'zawartosc', data_ważności);
```

Jak długo ma być ważne ciasteczko? Datę musimy podać jako liczbę sekund jaką upłynęła od początku roku 1970 roku do czasu kiedy ciasteczko ma wygasnąć.

Funkcja `time()` – zwróci aktualną datę (Patrz: https://www.w3schools.com/php/func_date_time.asp), dodajemy do niej tyle sekund ile chcemy by było ważne ciasteczko

```
setcookie('nick', 'Kasia', time()+60*60);  
setcookie('nick', 'Kasia', time()+3600);
```

Ciastko będzie ważne godzinę.

Metodą nie tyle przekazywania parametrów, co przechowywania niewielkich ilości danych na komputerze oglądającego stronę (np. informacje o imieniu i nazwisku lub nazwie użytkownika w tym serwisie) jest mechanizm cookies (ciasteczka).

Ciasteczka ustawione przez dany serwis dostępne są tylko dla niego i ustawiane są na jakiś czas, nie może być większe niż 4096 bajtów (4 kb).

Wysyłanie ciasteczka do przeglądarki internetowej to operacja niewidoczna dla użytkownika.

Cookies są przechowywane przez przeglądarkę w plikach tymczasowych.

Tablica \$_COOKIE

Wyświetlenie zawartości ciasteczka

Dane są przechowywane w tablicy ciasteczek \$_COOKIE (tak jak dane get są przechowywane w tablicy \$_GET).

Tablica \$_COOKIE zawiera elementy o indeksach takich jak nazwa ciastka (czyli w tym przypadku "nick").

```
echo "Masz na imię ".$_COOKIE['nick'];
```

Kilka informacji o cookies

Ciastka wysyłamy jako jeden z pierwszych elementów na stronie, żaden tekst nie może zostać wysłany do użytkownika przed ciasteczkiem, ponieważ funkcja setcookie() zmienia nagłówki zwracane z serwera. Nagłówki muszą być wysłane przed tekstem właściwym.

Cookies są kodowane w formie tekstowej, przesyłając dane binarne, lub zawierające znaki specjalne, ciasteczko należy zakodować.

Do kodowania można użyć algorytmu base64.

```
setcookie('nick', base64_encode('Kasia'), time()+3600);  
echo "Masz na imię ".base64_decode($_COOKIE['nick']);
```

Instrukcja setcookie()

```
setcookie($nazwa, $wartosc, $koniec, $sciezka,  
         $domena, $bezpieczne);
```

Setcookie() może przyjąć 6 argumentów. Jednak wymagane jest podanie tylko pierwszego, którym jest nazwa ciasteczka.

\$wartosc to wartość przypisana do ciastka,

\$koniec wyraża datę wygaśnięcia,

\$sciezka i \$domena mogą być stosowane do określenia adresów, dla których cookie jest ważne.

Ostatni argument, \$bezpieczne, oznacza, że cookie nie będzie wysyłane przez zwykłe połączenie HTTP.

Zastosowanie cookies

W praktyce, najczęściej używamy trzech pierwszych argumentów.

Wszelkie zapamiętywanie zalogowanego użytkownika, oddania głosu w ankiecie itp. są obsługiwane właśnie z wykorzystaniem cookies.

Jeśli cookie istnieje, to znaczy, że głos został oddany. Jeżeli nie, można oddać głos, po czym cookie jest tworzone.

```
// zapisanie oddania głosu jednorazowego
setcookie('oddano_glos', '1');

// w przypadku gdy głosować można raz dziennie
setcookie('oddano_glos', '1', time()+3600*24);
```

Ostatnie odwiedziny

```
<?php
if(!isset($_COOKIE['wizyta']))
{
    setcookie('wizyta', time(), time()+30*86400);
    echo 'Witaj, gościu.';
}
else
{
    setcookie('wizyta', time(), time()+30*86400);
    echo 'Witaj, ostatni raz odwiedziłeś nas '
        .date('d.m.Y, H:i', $_COOKIE['wizyta']);
}
```

Wynik:

Witaj, ostatni raz odwiedziłeś nas 21.02.2016, 13:11

W skrypcie sprawdzamy, czy ustawialiśmy już ciastko dla danego internauty.

Jeśli nie, tworzymy je i wyświetlamy komunikat powitania. W przeciwnym wypadku także aktualizujemy wartość, ale też wyświetlamy datę ostatniej wizyty odczytaną właśnie z ciastka.

Przy pierwszym wejściu umieścić na np. miesiąc ciastko z datą ostatniej wizyty, po czym ją sukcesywnie odczytywać

Wywołanie setcookie() nie nadpisuje wartości w tablicy \$_COOKIE. Dlatego bez problemu można uprościć skrypt:

```
<?php
setcookie('wizyta', time(), time()+30*86400);
if(!isset($_COOKIE['wizyta']))
{
    echo 'Witaj, gościu.';
}
else
{
    echo 'Witaj, ostatni raz odwiedziłeś nas '
        .date('d.m.Y, H:i', $_COOKIE['wizyta']);
}
```

Skrypt informujący o ostatniej wizycie na stronie

Aby istniejące ciastko skasować, wywołujemy funkcję `setcookie()` z jakąś przeszłą datą ważności

```
<?php
    setcookie('wizyta', '', 0);
?>
```

Przechowywanie tablic

Standardowo nie ma możliwości przechowywania tablic w ciasteczkach, lecz można to zrobić "na około".

Przed zapisaniem tablicy w ciasteczku należy użyć funkcji `serialize()`.

Zamienia ona tablicę na ciąg znaków (Patrz: https://www.w3schools.com/php/func_var_serialize.asp).

Do ponownego odczytania takiej tablicy służy funkcja `unserialize()`

(Patrz: https://www.w3schools.com/php/func_var_unserialize.asp).

```
<?php
// zapis
$tablica = Array('a' => 'pierwszy', 'b' => 'drugi');
setcookie('tablica', serialize($tablica), time()+3600);

//odczyt zabezpieczony przed nieistniejącym ciasteczkiem
if (isset($_COOKIE['tablica']))
{
    $tablica = unserialize($_COOKIE['tablica']);
}
else
{
    $tablica = Array();
}
```

Licznik wejść

```
<?php
if(!isset($_COOKIE['licznik']))
{
    $wygasniecie = time() + 60 * 60 * 24;
    setcookie('licznik', 'odwiedzil', $wygasniecie);

    $fp = fopen('wejscia.txt', 'r');
    $ile = intval(fgets($fp, 100));
    fclose($fp);
    $fp = fopen('wejscia.txt', 'w');
    $ile = $ile + 1;
    fwrite($fp, $ile);
    fclose($fp);
}
$fp = fopen('wejscia.txt', 'r');
$ile = fgets($fp, 100);
echo 'Jestes '.$ile.' odwiedzajacym ta witryne';
fclose($fp);
/* Należy stworzyć wejscia.txt,
wpisać do niego liczbę i zapisać */
```

Przykład przesyłania danych użytkownika czytanych z formularza.

Dane użytkownika zostaną przekazane przez formularz, który będzie zawierał pole do wprowadzania imienia i nazwiska użytkownika. Pobrany z formularza parametr zawierający nazwisko i imię użytkownika zostanie przekazany do utworzonego pliku cookie.

```
<?php
    if (!isset($_COOKIE['dane']) && !isset($_POST['nazw'])) {
?>

<!DOCTYPE HTML>

<html>

<head>

<title>Dane_user</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>

<body>

<form action="http://localhost/rejestracja.php" method="post">
Podaj nazwisko i imię:<br/>
<input type="text" name="nazw" value=" " size="35">
<p><input type="submit" value="Wyślij" name="wyslij"></p>
</form>

<?php
}
else {
    if (isset($_POST['nazw'])) {
        setcookie ('dane', $_POST['nazw'], time()+60*60*24*365);
        echo "<p>Dziękujemy za wprowadzenie danych.</p>";
    }
    else
        echo "<p>Witamy po raz kolejny! ".$_COOKIE['dane']."</p>";
}
?>

</body>

</html>
```

W powyższym skrypcie rozpatrywane są trzy sytuacje:

- W pierwszej plik cookie nie został utworzony i z formularza nie zostały przesłane dane — wtedy należy wywołać formularz rejestracyjny.
- W drugiej dane z formularza zostały przesłane, ale nie istnieje plik cookie — wtedy należy utworzyć taki plik.

- W trzeciej dane z formularza zostały przesłane oraz istnieje plik cookie — wtedy wyświetli się komunikat witający użytkownika po raz kolejny na stronie.

Zapisana w kodzie instrukcja `if(!isset($_COOKIE['dane']) && !isset($_POST['nazw']))` sprawdza, czy istnieje plik cookie i czy z formularza został przekazany parametr zawierający nazwisko i imię użytkownika. Jeżeli nie, to zostanie utworzona strona zawierająca formularz. Z formularza za pomocą metody POST zostanie przekazany parametr `nazw`.

Druga instrukcja `if(isset($_POST['nazw']))` sprawdza, czy do skryptu zostały przesłane dane z formularza. Jeżeli tak, to za pomocą funkcji `setcookie()` do przeglądarki jest wysyłany plik cookie o nazwie `dane`. Jego wartością jest wartość parametru `nazw` pobrana z tablicy `$_POST`. Czas ważności pliku został ustawiony na 236 dni. Na stronie wyświetli się tekst z podziękowaniem za wprowadzenie danych.

Trzecia sytuacja wystąpi, jeśli istnieje plik cookie. Wtedy na stronie wyświetli się informacja o rozpoznaniu użytkownika.