

Obsługa ciasteczek w JS

Dzięki ciasteczkom obsługiwany w JS za pomocą

`document.cookie` przeglądarka zapamiętuje wybrane przez nas dane lokalnie na maszynie klienta – nawet jeśli zamknie stronę i wróci do niej po pewnym czasie. Możemy przechowywać w nich np. różne ustawienia użytkownika, jego udzielone zgody czy preferencje wyświetlania strony (np. ciemny *theme* zamiast domyślnego, jasnego). To potężne narzędzie – dlatego warto je dobrze zrozumieć. Również na egzaminie INF.03 zdarzają się zadania wymagające od nas zapisywania, bądź odczytu informacji przechowywanych w ciasteczkach.

Co to są ciasteczka (cookies)?

Ciasteczko to mały fragment danych, który przeglądarka przechowuje i wysyła do serwera przy każdej wizycie na stronie. Kiedyś definiowaliśmy je jako małe pliki tekstowe zapisane na komputerze klienta, zaś we współczesnych przeglądarkach są to często rekordy przechowywane w lokalnej bazie danych SQLite przeglądarki. Złota zasada pozostaje jednak zawsze ta sama – ciasteczka są przechowywane lokalnie, na komputerze klienckim, a nie na serwerze.

Tworzenie ciasteczek w JavaScript

Rozważmy prosty przykład – po kliknięciu na przycisk zapiszemy ciasteczko z informacją, że odwiedziliśmy stronę i poinformujemy o tym fakcie użytkownika:

```
<div id="wynik"></div>
<button onclick="ustawCiasteczko()">Ustaw ciasteczko</button>
<script>
  function ustawCiasteczko() {
    document.cookie = "odwiedzono=1; expires=Fri, 31 Dec 2025 23:59:59 GMT;
path=/";
    document.getElementById("wynik").innerHTML = "Ciasteczko zostało zapisane!";
  }
</script>
```

Co robi ten kod? Pomysł polega na tym, że po kliknięciu przycisku:

- W przeglądarce zapisuje się ciasteczko o nazwie `odwiedzono` i wartości `1`,
- Następnie w pustym bloku `<div id="wynik">` pojawia się tekst informujący o tym fakcie.

Wyjaśnienie poszczególnych elementów kodu:

- Blok `<div id="wynik"></div>` to pusty kontener, w którym po kliknięciu pojawi się komunikat.
- Tagi `<button onclick="ustawCiasteczko()">...</button>` definiują przycisk, który po kliknięciu wywołuje funkcję JavaScript.
- `document.cookie = "odwiedzono=1; ..."` – linijka odpowiedzialna za ustawienie ciasteczka:
 - `expires=...` – data “wygaśnięcia” ciasteczka (tutaj: 31.12.2025 o 23:59:59 GMT).
 - `path=/` – ciasteczko będzie dostępne w całym serwisie (nie tylko na jednej podstronie).
 - `document.getElementById("wynik").innerHTML = ...` – aktualizujemy treść w bloku `div`.

Do zapamiętania:

- Ustawianie ciasteczek w JS odbywa się poprzez przypisanie odpowiedniego tekstu do `document.cookie`.
- Ciasteczka można ograniczyć czasowo, określając kiedy “wygasają” (*ang. expires*) oraz przypisać do konkretnej ścieżki (*ang. path*) w serwisie.
- Możemy reagować na kliknięcie przycisku poprzez prosty atrybut `onclick` lub lepiej – przez `addEventListener` – szczegóły w lekcji: [Atrybut onclick vs. addEventListener](#).
- Funkcje takie jak ta są bardzo przydatne do tworzenia powiadomień, pamiętania preferencji użytkownika, itd. choć oczywiście nie zawsze będziemy tak wprost informować użytkownika komunikatem o zapisaniu ciasteczka. Najważniejsze to przyswoić sobie sam akt tworzenia ciasteczka, jako przypisanie odpowiedniego tekstu do `document.cookie`.

Odczytywanie ciasteczek w JavaScript

Teraz zajmiemy się odczytywaniem wartości ciasteczek, które wcześniej zostały zapisane w przeglądarce użytkownika. To bardzo przydatna umiejętność, jeśli chcemy np. sprawdzić, czy ktoś odwiedził już naszą stronę. Przykład ten będzie powiązany z poprzednią funkcją ustawiającą ciasteczko z wartością

```
odwiedzono=1
```

Po kliknięciu przycisku poniżej, przeglądarka spróbuje odczytać wartość ciasteczka o nazwie

`odwiedzono` i wyświetli ją na stronie. Użyjemy do tego funkcji pomocniczej `pobierzCiasteczko()`:

```
<div id="wynik"></div>
<button onclick="pokazCiasteczko()">Czy już odwiedzono?</button>
<script>
  function pobierzCiasteczko(nazwa) {
    let ciasteczka = document.cookie.split("; ");
    for (let i = 0; i < ciasteczka.length; i++) {
      let para = ciasteczka[i].split("=");
      if (para[0] === nazwa) {
        return para[1];
      }
    }
    return null;
  }
  function pokazCiasteczko() {
    let odwiedzono = pobierzCiasteczko("odwiedzono");
    let komunikat = (odwiedzono === "1")
      ? "To nie jest Twoja pierwsza wizyta!"
      : "Wygląda na to, że jesteś tu pierwszy raz lub ciasteczko wygasło.";
    document.getElementById("wynik").innerHTML = komunikat;
  }
</script>
```

Co się wydarzyło w tym skrypcie?

- Pojemnik `<div id="wynik"></div>` – to oczywiście miejsce, gdzie zostanie wyświetlony komunikat zależny od tego, czy ciasteczko istnieje.
- Przycisk `onclick="pokazCiasteczko()"` – jego kliknięcie wywołuje funkcję sprawdzającą istnienie ciasteczka.
- Funkcja `pobierzCiasteczko(nazwa)`:

- pobiera ciasteczka jako ciąg znaków z `document.cookie`,
 - rozdziela je na pary `nazwa=wartość`,
 - sprawdza każdą parę, szukając tej o nazwie `odwiedzono`,
 - zwraca jej wartość (lub `null`, jeśli jej nie ma).
- Funkcja `pokazCiasteczko()` w zależności od wartości ciasteczka pokazuje odpowiedni komunikat. Na podstawie istnienia (bądź nie) ciasteczka możemy stwierdzić, czy użytkownik był już na stronie – to świetna okazja, by np. wczytać personalizację treści.
 - Możemy łatwo rozszerzyć taki mechanizm o więcej ciasteczek, np. imię, konkretną datę ostatniej wizyty, preferencje wyglądu itp. zależnie od naszych intencji.

Usuwanie ciasteczek w JavaScript

Ciasteczka, które raz zapiszemy, nie zostaną automatycznie usunięte – musimy o to zadbać sami, jeśli chcemy je “wyczyścić” przed wygaśnięciem ustawionego czasu ich życia. W JavaScript możemy to zrobić na dwa główne sposoby: ustawiając ciasteczku datę ważności w przeszłości albo nadając mu bardzo krótki czas życia. Pokażmy obie metody na przykładzie ciasteczka

`odwiedzono`, które we wcześniejszych przykładach zapisaliśmy.

Metoda 1 – ustawienie daty w przeszłości

Najczęstsza w praktyce i najbardziej pewna metoda. Ustawiamy ciasteczku taką samą nazwę i ścieżkę jak wcześniej, ale z datą ważności, która już minęła – dzięki temu przeglądarka je od razu usuwa.

```
<button onclick="usunCiasteczko()">Usuń ciasteczko</button>
<script>
function usunCiasteczko() {
    document.cookie = "odwiedzono=; expires=Thu, 01 Jan 1970 00:00:00 GMT; path=/";
    document.getElementById("wynik").innerHTML = "Ciasteczko zostało usunięte.";
}
</script>
```

Ustawiamy wartość pustą (choć nie jest to obowiązkowe) oraz bardzo starą datę – 1 stycznia 1970 to jak być może wiemy rozpoczęcie tzw. “epoki UNIX” (standard POSIX), czyli początek zliczania czasu w informatyce. Oczywiście data ta nie musi być w praktyce aż tak “stara” – wystarczy jakakolwiek “przeszła” wartość.

Metoda 2 – bardzo krótki czas życia

Możemy też ustawić ciasteczko z krótkim “czasem życia” – np. kilka sekund. Po tym czasie przeglądarka automatycznie je usunie.

```
<button onclick="ustawTymczasoweCiasteczko()">Ustaw ciasteczko na 10 sekund</button>
<script>
function ustawTymczasoweCiasteczko() {
    let teraz = new Date();
    teraz.setSeconds(teraz.getSeconds() + 10); // dodajemy 10 sekund
    document.cookie = "odwiedzono=1; expires=" + teraz.toUTCString() + "; path=/";
    document.getElementById("wynik").innerHTML = "Ustawiono tymczasowe ciasteczko (10 s).";
}
</script>
```

To rozwiązanie jest przydatne, jeśli chcemy np. przechować informację tylko na czas jednej sesji lub ograniczonego, roboczego testu.

Wnioski na temat usuwania ciasteczek:

- Ciasteczka możemy usuwać, nadpisując je tą samą nazwą, ale z przeszłą datą wygaśnięcia.
- Alternatywnie – możemy nadać im bardzo krótki czas życia, jeśli mają zniknąć automatycznie.
- Pamiętajmy, by zawsze używać tej samej `path`, co przy ustawianiu ciasteczka – inaczej przeglądarka nie nadpisze właściwego ciasteczka!

Dzięki tym metodom mamy pełną kontrolę nad tym, co i jak długo przechowujemy po stronie klienta.

Podsumowanie – co zapamiętać z lekcji?

- Ciasteczka w JavaScript przechowujemy lokalnie w przeglądarce przy użyciu `document.cookie`.
- Umożliwiają one zapamiętywanie ustawień użytkownika, preferencji lub np. informacji o odwiedzinach strony.
- Do ustawienia ciasteczka wystarczy przypisanie ciągu tekstowego z nazwą, wartością, datą wygaśnięcia `expires` oraz ścieżką `path`.
- Do odczytu ciasteczka przeszukujemy `document.cookie` i dzielimy go na pary `nazwa=wartość`.
- Ciasteczko można usunąć, ustawiając mu datę wygaśnięcia w przeszłości.
- Alternatywnie, możemy ustawić ciasteczko z krótkim czasem życia, np. 10 sekund.
- Do dynamicznego reagowania na zdarzenia (np. kliknięcia) możemy używać `onclick` lub `addEventListener`.
- Mechanizmy te są przydatne np. przy personalizacji treści, wyświetlaniu komunikatów powitalnych, lub pamiętaniu wyborów użytkownika.