

Tabela 4.1. Wybrane właściwości obiektu window

Nazwa	Znaczenie	Dostępność
closed	Ustawiona na <code>true</code> oznacza, że okno zostało zamknięte.	FF, IE, NN, OP, W3C
defaultStatus	Domyślny tekst wyświetlany na pasku stanu.	FF, IE, NN, OP, W3C
document	Obiekt <code>document</code> zawierający elementy wyświetlanej witryny.	FF, IE, NN, OP, W3C
event	Zawiera odniesienie do obiektu typu <code>Event</code> opisującego ostatnie zdarzenie. Używany w modelu zdarzeń przeglądarki Internet Explorer.	IE, OP
frames	Tablica zawierająca odniesienia do ramek danego okna.	FF, IE, NN, OP
history	Obiekt <code>History</code> zawierający historię odwiedzin.	FF, IE, NN, OP, W3C
innerWidth	Wewnętrzna wysokość obszaru okna.	NN, OP, FF
innerHeight	Wewnętrzna szerokość obszaru okna.	NN, OP, FF
length	Liczba ramek potomnych zawartych w oknie.	FF, IE, NN, OP, W3C
location	Obiekt <code>Location</code> zawierający URL aktualnego dokumentu.	FF, IE, NN, OP, W3C
locationbar	Obiekt pozwalający na określanie, czy ma być widoczny pasek adresu. Posiada właściwość <code>visible</code> , której można przypisywać wartości <code>true</code> i <code>false</code> .	NN, FF
menubar	Obiekt pozwalający na określanie, czy ma być widoczny pasek menu. Posiada właściwość <code>visible</code> , której można przypisywać wartości <code>true</code> i <code>false</code> .	NN, FF

name	Nazwa bieżącego okna.	FF, IE, NN, OP, W3C
opener	Jeżeli bieżące okno zostało otwarte za pomocą metody <code>open</code> , właściwość <code>opener</code> zawiera odniesienie do okna źródłowego.	NN, OP, FF, W3C
outerHeight	Zewnętrzna wysokość obszaru okna.	NN, OP, FF, W3C
outerWidth	Zewnętrzna szerokość obszaru okna.	NN, OP, FF, W3C
pageXOffset	Przesunięcie bieżącej strony w poziomie względem lewego górnego rogu okna (szerokość zasłoniętego obszaru).	NN, OP, FF, W3C
pageYOffset	Przesunięcie bieżącej strony w pionie względem lewego górnego rogu okna (wysokość zasłoniętego obszaru).	NN, OP, FF, W3C
parent	Odniesienie do okna źródłowego.	FF, IE, NN, OP, W3C
personalbar	Obiekt pozwalający na określanie, czy ma być widoczny pasek ustawień osobistych. Posiada właściwość <code>visible</code> , której można przypisywać wartości <code>true</code> i <code>false</code> .	NN, FF
scrollbars	Obiekt pozwalający na określanie, czy mają być widoczne paski przewijania. Posiada właściwość <code>visible</code> , której można przypisywać wartości <code>true</code> i <code>false</code> .	NN, FF
self	Odniesienie do bieżącego aktywnego okna lub ramki.	FF, IE, NN, OP, W3C
status	Tekst wyświetlany na pasku stanu.	FF, IE, NN, OP, W3C
statusbar	Obiekt pozwalający na określanie, czy ma być widoczny pasek stanu. Posiada właściwość <code>visible</code> , której można przypisywać wartości <code>true</code> i <code>false</code> .	NN, FF

toolbar	Obiekt pozwalający na określanie, czy ma być widoczny pasek narzędziowy. Posiada właściwość <code>visible</code> , której można przypisywać wartości <code>true</code> i <code>false</code> .	NN, FF
top	Odniesienie do okna znajdującego się najwyżej w hierarchii.	FF, IE, NN, OP, W3C
window	Odniesienie do bieżącego okna lub ramki	FF, IE, NN, OP

Lista Z Zapisywaniem Imion

Ten fragment kodu JavaScript obsługuje przycisk o identyfikatorze "addNameButton", który znajduje się na stronie. Kod ten wykonuje następujące czynności:

Dodaje "słuchacza zdarzeń" (event listener): `addNameButton.addEventListener('click', () => { ... });`

To oznacza, że kiedy użytkownik kliknie na przycisk o identyfikatorze "addNameButton", zostanie wykonana określona funkcja obsługi zdarzeń.

Wewnątrz funkcji obsługi zdarzenia (czyli tego, co dzieje się po kliknięciu przycisku), znajdują się następujące kroki:

`const name = nameInput.value;` Pobiera wartość wprowadzoną przez użytkownika do pola tekstowego o identyfikatorze "nameInput" i przypisuje ją do zmiennej "name". To jest imię, które użytkownik wpisuje w polu tekstowym.

`if (name) { ... }` Sprawdza, czy zmienna "name" nie jest pusta. Oznacza to, że użytkownik coś wpisał do pola tekstowego. Warunek ten sprawdza, czy "name" zawiera jakąkolwiek zawartość (nie jest puste).

Wewnątrz bloku "if" znajdują się następujące czynności:

`const names = JSON.parse(window.localStorage.getItem('names')) || [];` Pobiera dane z lokalnego magazynu przeglądarki o kluczu "names" (jeśli już istnieją) i parsuje je jako obiekt JavaScript. Jeśli dane nie istnieją, zostanie utworzona pusta tablica (jeśli "names" nie istnieje w magazynie przeglądarki).

`names.push(name);` Dodaje imię (wartość zmiennej "name") do tablicy "names". Oznacza to, że nowe imię jest dodawane do istniejących imion w magazynie przeglądarki.

`window.localStorage.setItem('names', JSON.stringify(names));` Zapisuje zaktualizowaną tablicę "names" do lokalnego magazynu przeglądarki pod kluczem "names". Dane są zapisywane w magazynie przeglądarki jako ciąg znaków JSON.

`nameInput.value = ";;` Czyści pole tekstowe "nameInput", aby użytkownik mógł wprowadzić nowe imię.

`refreshNameList();` Wywołuje funkcję "refreshNameList()" w celu odświeżenia listy imion na stronie.

```
const names = JSON.parse(window.localStorage.getItem('names')) || [];
```

Pobiera dane z lokalnego magazynu przeglądarki o kluczu "names". Wartość ta jest zapisana w magazynie przeglądarki jako ciąg znaków JSON. Funkcja `JSON.parse()` służy do przekształcenia ciągu JSON z magazynu w obiekt JavaScript. Jeśli dane nie istnieją lub nie można ich sparsować, zostaje utworzona pusta tablica.

```
nameList.innerHTML = ";;
```

Czyści zawartość elementu o identyfikatorze "nameList". To jest element `` (lista nieuporządkowana), w którym wyświetlane są imiona użytkowników.

```
names.forEach(name => { ... });;
```

Wykorzystuje pętlę `forEach()` do iteracji przez tablicę "names". W każdej iteracji wykonuje następujące operacje:

a. `const listItem = window.document.createElement("li");`

Tworzy nowy element listy ``. Jest to element, który zostanie dodany do listy imion.

b. `listItem.textContent = name;`

Ustawia tekst elementu `` na imię, które zostało pobrane z tablicy "names".

c. `nameList.appendChild(listItem);`

Dodaje nowo utworzony element `` (reprezentujący imię) do listy imion (elementu `` o identyfikatorze "nameList"). Dzięki temu każde imię jest reprezentowane jako nowy element na liście.

Zegar

```
const timer = setInterval(() => { ... }, 1000);:
```

setInterval() to funkcja wbudowana w JavaScript, która wykonuje określone działania (czyli kod zawarty w funkcji) co określony okres czasu. W tym przypadku okres czasu wynosi 1000 milisekund, co odpowiada 1 sekundzie.

Kod wewnątrz setInterval() jest wykonywany cyklicznie co 1 sekundę.

Wewnątrz funkcji strzałkowej () => { ... }, znajdują się następujące kroki:

a. elapsedSeconds++;:

Zwiększa wartość zmiennej elapsedSeconds o 1 za każdym razem, gdy kod wewnątrz interwału zostaje wykonany. Ta zmienna służy do śledzenia upływającego czasu w sekundach.

b. elapsedTimeElement.textContent = `\${elapsedSeconds} sekund;:

Aktualizuje zawartość elementu HTML o identyfikatorze "elapsedTime" (zdefiniowanego wcześniej w kodzie HTML) za pomocą właściwości textContent. Wartość przypisywana do textContent to łańcuch znaków, który zawiera aktualną liczbę sekund zmienną elapsedSeconds.

Dzięki tej linii kodu, na stronie wyświetlany jest bieżący czas spędzony na stronie, który jest zwiększany co sekundę.

Źródło strony

window.location.href:

window.location to obiekt dostarczany przez przeglądarkę, który zawiera informacje o bieżącym adresie URL strony.

Wartość href jest jednym z właściwości obiektu window.location i przechowuje pełny adres URL aktualnie wyświetlanej strony.

currentURL.textContent = window.location.href;:

Pobiera pełny adres URL bieżącej strony, który znajduje się w właściwości window.location.href.

Następnie przypisuje ten adres URL do właściwości textContent elementu o identyfikatorze "currentURL".

W efekcie, na stronie internetowej, w miejscu, gdzie znajduje się element o identyfikatorze "currentURL", zostanie wyświetlony aktualny adres URL.

Ciągi znaków (łańcuchy, stringi) - dowolny ciąg znaków objęty cudzysłowem może być wartością w formacie JSON, na przykład

"To jest poprawny łańcuch znaków" . Znaki cudzysłowia okalające łańcuch muszą być podwójne (znak ") - pojedyncze (znak ' są niepoprawne!).

window.document: Reprezentuje aktualny dokument HTML na stronie.

window.location: Udostępnia informacje o bieżącym adresie URL.

window.navigator: Udostępnia informacje o przeglądarce użytkownika.

window.history: Pozwala na zarządzanie historią przeglądania.

window.localStorage i **window.sessionStorage**: Udostępniają mechanizmy przechowywania danych na stronie.

window.alert(), **window.confirm()**, **window.prompt()**: Służą do interakcji z użytkownikiem poprzez wyświetlanie okienek dialogowych.

window.setTimeout() i **window.setInterval()**: Służą do planowania operacji asynchronicznych. Asynchroniczny = nie dzieje się w tym samym czasie

window.fetch(): Pozwala na wysyłanie żądań HTTP.

window.XMLHttpRequest: Starszy sposób wykonywania żądań HTTP.

window.console: Udostępnia konsolę przeglądarki do logowania informacji.

window.innerWidth i **window.innerHeight**: Zwracają szerokość i wysokość okna przeglądarki.

window.screen: Udostępnia informacje o ekranie użytkownika.

window.localStorage: Pozwala na przechowywanie danych w pamięci lokalnej przeglądarki.

window.sessionStorage: Pozwala na przechowywanie danych w pamięci sesji przeglądarki.

window.parent i **window.frames**: Służą do nawigacji w ramkach i oknach przeglądarki.

