**LocalStorage**

1. **Zapisywanie i odczytywanie wartości z LocalStorage**

* Napisz program, który pozwoli użytkownikowi wprowadzić swoje imię i zapisze je w LocalStorage. Osobnym przyciskiem użytkownik może odczytać zapisane imię z LocalStorage i wyświetlić je na ekranie.
* Aby korzystać z LocalStorage, można użyć metod localStorage.setItem(key, value) do zapisywania wartości oraz localStorage.getItem(key) do odczytywania wartości.

1. **Rzut kostką i LocalStorage**

* Napisz program, który pozwala na wykonanie rzutu kostką sześciościenną i wyświetla wynik na ekranie. Ostatnio wyrzucona wartość powinna być zapamiętywana również po zamknięciu i ponownym otwarciu strony.
* Zmodyfikuj program, aby zapamiętywał ostatnie 10 rzutów.
* LocalStorage przechowuje dane w postaci tekstowej, więc jeśli potrzebujemy przechowywać obiekty JavaScript (np. tablicę wylosowanych wartości), musimy je najpierw przekonwertować na tekst (np. za pomocą JSON.stringify()) i przy odczycie przekonwertować z powrotem na obiekt (np. za pomocą JSON.parse()).

1. **Zapamiętywanie preferencji użytkownika**
   * + Napisz program, który pozwala użytkownikowi zdefiniować swoje preferencje, takie jak kolor tła strony, kolor tekstu i rozmiar czcionki. Program powinien zapisać te preferencje w LocalStorage, aby móc je przywrócić przy kolejnym odwiedzeniu strony.
2. **Koszyk zakupowy w LocalStorage (\*\*\*)**

* Napisz program, który symuluje koszyk zakupowy. Użytkownik może dodawać produkty do koszyka, a program będzie przechowywać te informacje w LocalStorage. Program powinien umożliwiać dodawanie, usuwanie i wyświetlanie produktów w koszyku.

**Timery**

1. **SetTimeout**

* Przetestuj dziełanie funkcji setTimeout pisząc program, który po naciśnięciu przycisku czeka dwie sekundy i wyświetla komunikat na ekranie.

1. **SetInterval**

* Przetestuj działanie funkcji setInterval pisząc program, który wyświetla zegar. Aktualny czas możesz sprawdzić przy pomocy obiektu Date. Funkcja setInterval pozwoli odświeżać wartość zegara co sekundę.

1. **Stoper**

* Napisz program, który wykorzystuje funkcję setInterval, aby wyświetlać na ekranie działający stoper, odmierzający czas z dokładnością do dziesiątych sekundy. Jeden przycisk pozwala na uruchomienie stopera (odliczanie wartości rozpoczyna się od 0), a na jego zatrzymanie (przy pomocy funkcji clearInterval)
* Dodaj trzeci przycisk, który pozwoli na notowanie „międzyczasów” – zapamiętuje i wypisuje na ekranie aktualną wartość stopera, ale bez jego zatrzymywania.

1. **Minutnik**

* Napisz program, który pozwoli na realizację prostego minutnika. Użytkownik wybiera liczbę minut i sekund do odmierzenia, po czym uruchamia przyciskiem minutnik. Czas jest odliczany „w dół” i na bieżąco wyświetlany na ekranie. Po dobięgnięciu licznika do 0, wyświetlany jest komunikat a ekran zmienia kolor.

1. **Animacja przesuwającego się elementu**

* Stwórz prostą animację, w której element na stronie przesuwa się w prawo z określoną prędkością. Wykorzystaj funkcję setInterval do cyklicznego aktualizowania pozycji elementu na stronie. Jeśli nadasz elementowi wartość stylu „position: aboslute”, możesz ustwić jego pozycję przy pomocy właściwości stylu „left”.

1. **Powiadomienia na żywo (\*\*\*)**

* Napisz program, który symuluje powiadomienia na żywo. Co losową liczbę sekund na ekranie pojawia się nowe powiadomienie, które przez pewien czas pozostaje wyświetlone, a następnie zniknie. Wykorzystaj funkcji setTimeout lub/i setInterval do generowania nowych powiadomień i zarządzania ich widocznością na ekranie.