Zdarzenia w JavaScript

1. Wprowadzenie do zdarzeń w JavaScript

Zdarzenia w JavaScript to akcje zachodzące na stronie internetowej, na które można reagować za pomocą kodu JavaScript. Mogą to być kliknięcia myszką, naciśnięcia klawiszy, załadowanie strony czy przesłanie formularza.

Jak obsługiwać zdarzenia?

Aby obsłużyć zdarzenie, można użyć jednej z trzech metod:

1. **Atrybuty HTML** (np. `onclick`, `onmouseover`)
2. **Metoda **``
3. **Przypisanie funkcji do właściwości zdarzenia**

2. Przykłady obsługi zdarzeń

2.1. Obsługa zdarzeń za pomocą atrybutów HTML

```
<button onclick="alert('Kliknięto!')">Kliknij mnie</button>
```

To podejście nie jest zalecane, ponieważ utrudnia zarządzanie kodem JavaScript.

2.2. Obsługa zdarzeń za pomocą `addEventListener`

```
<button id="myButton">Kliknij mnie</button>
<script>
  document.getElementById("myButton").addEventListener("click", function() {
    alert("Przycisk kliknięty!");
  });
</script>
```

To bardziej elastyczne podejście, ponieważ umożliwia dodanie wielu obsług zdarzeń.

2.3. Obsługa zdarzeń poprzez przypisanie funkcji do właściwości zdarzenia

```
<button id="myButton">Kliknij mnie</button>
<script>
  let button = document.getElementById("myButton");
  button.onclick = function() {
    alert("Przycisk kliknięty!");
  };
</script>
```

3. Popularne zdarzenia w JavaScript

3.1. Zdarzenia myszy

Zdarzenie	Opis
`click`	Kliknięcie myszą
`dblclick`	Podwójne kliknięcie
`mousedown`	Wciśnięcie przycisku myszy
`mouseup`	Zwolnienie przycisku myszy
`mousemove`	Ruch myszą
`mouseover`	Najechanie na element
`mouseout`	Opuszczenie elementu

Przykład `mouseover` i `mouseout`

```
<div id="box" style="width:100px; height:100px; background-color:blue;"></div>
<script>
  let box = document.getElementById("box");
  box.addEventListener("mouseover", function() {
    box.style.backgroundColor = "red";
  });
  box.addEventListener("mouseout", function() {
    box.style.backgroundColor = "blue";
  });
</script>
```

3.2. Zdarzenia klawiatury

Zdarzenie	Opis
`keydown`	Naciśnięcie klawisza
`keyup`	Zwolnienie klawisza
`keypress`	Naciśnięcie klawisza (przestarzałe)

Przykład `keydown`:

```
<input type="text" id="myInput">
<script>
  document.getElementById("myInput").addEventListener("keydown", function(event) {
    console.log("Naciśnięto klawisz: " + event.key);
  });
</script>
```

3.3. Zdarzenia formularzy

Zdarzenie	Opis
`submit`	Przesłanie formularza
`change`	Zmiana wartości pola

`focus`	Ustawienie kursora na polu
`blur`	Opuszczenie pola

Przykład `submit`:

```
<form id="myForm">
  <input type="text" name="username">
  <button type="submit">Wyślij</button>
</form>
<script>
  document.getElementById("myForm").addEventListener("submit", function(event) {
    event.preventDefault(); // Zapobiega domyślnemu wysłaniu formularza
    alert("Formularz wysłany!");
  });
</script>
```

3.4. Zdarzenia okna

Zdarzenie	Opis
`load`	Załadowanie strony
`resize`	Zmiana rozmiaru okna
`scroll`	Przewijanie strony

Przykład `scroll`:

```
<script>
  window.addEventListener("scroll", function() {
    console.log("Przewijasz stronę!");
  });
</script>
```

4. Delegowanie zdarzeń

Zamiast dodawać zdarzenie do każdego elementu osobno, można obsłużyć je na poziomie nadrzędnego elementu.

Przykład:

```
<ul id="myList">
  <li>Kliknij mnie</li>
  <li>Kliknij mnie</li>
  <li>Kliknij mnie</li>
</ul>
<script>
  document.getElementById("myList").addEventListener("click", function(event) {
    if (event.target.tagName === "LI") {
      alert("Kliknięto: " + event.target.textContent);
    }
  });
</script>
```

5. Podsumowanie

- JavaScript obsługuje wiele zdarzeń związanych z myszą, klawiaturą, formularzami i oknem przeglądarki.
- `addEventListener` jest najczęściej używaną metodą obsługi zdarzeń.
- Delegowanie zdarzeń pozwala optymalizować kod dla dynamicznych elementów strony.

CZĘŚĆ 2

1. Odczytywanie danych z formularza

Aby odczytać dane wpisane przez użytkownika do formularza, używamy `document.getElementById()` lub `querySelector()`.

Przykład: Pobieranie danych z formularza

```html
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Odczyt danych z formularza</title>
</head>
<body>
    <form id="myForm">
        <label for="name">Imię:</label>
        <input type="text" id="name" name="name">
        <button type="button" onclick="readData()">Pokaż dane</button>
    </form>

    <p id="output"></p>

    <script>
        function readData() {
            let nameValue = document.getElementById("name").value;
            document.getElementById("output").innerText = "Wpisane imię: " + nameValue;
        }
    </script>
</body>
</html>
```

Jak to działa?

- Pobieramy wartość pola `input` przez `.value`.
- Wstawiamy ją do elementu `<p>` z `innerText`.

2. Wstawianie danych do formularza lub w inne miejsce

Dane można dynamicznie umieszczać w polach formularza oraz innych elementach strony.

Przykład: Wstawianie danych do formularza i nagłówka

```html
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Wstawianie danych</title>
</head>
<body>
    <input type="text" id="nameInput" placeholder="Wpisz swoje imię">
    <button onclick="insertData()">Zapisz</button>

    <h2 id="greeting"></h2>

    <script>
        function insertData() {
            let name = document.getElementById("nameInput").value;
            document.getElementById("greeting").innerText = "Witaj, " + name + "!";
            document.getElementById("nameInput").value = "Wpisano: " + name;
        }
    </script>
</body>
</html>
```

Co się dzieje?

- Po kliknięciu przycisku wartość z `input` trafia do nagłówka `<h2>`.
- Wartość w polu tekstowym też się zmienia.

3. Dynamiczna zmiana stylów CSS

Możemy zmieniać wygląd elementów w zależności od danych użytkownika.

Przykład: Zmiana koloru tła i rozmiaru tekstu

```html
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Zmiana CSS</title>
    <style>
        #box {
            width: 200px;
            height: 100px;
            background-color: lightgray;
            text-align: center;
            line-height: 100px;
        }
    </style>
</head>
<body>
    <div id="box">Tekst</div>
    <button onclick="changeStyle()">Zmień wygląd</button>

    <script>
        function changeStyle() {
            let box = document.getElementById("box");
            box.style.backgroundColor = "blue";
            box.style.color = "white";
            box.style.fontSize = "20px";
        }
    </script>
</body>
</html>
```

Efekt:
Po kliknięciu zmienia się kolor tła, kolor tekstu i jego rozmiar.

Zadania:

1. Pobierz wartość wpisaną w `input` po kliknięciu przycisku i wyświetl ją w `div`.
2. Po naciśnięciu klawisza w polu `input`, wyświetl aktualnie wpisany tekst w `p`.
3. Zmodyfikuj `input`, aby po kliknięciu przycisku jego wartość zmieniała się na `Hello World!`.
4. Po kliknięciu przycisku zmień wartość `input` na wartość domyślną (`placeholder`).
5. Po wpisaniu tekstu w `input` i kliknięciu przycisku dodaj ten tekst jako nowy `li` do listy `ul`.
6. Po kliknięciu przycisku zmień kolor tła strony na losowy kolor.
7. Po najechaniu myszką na `div`, zmień jego kolor na czerwony, a po opuszczeniu na zielony.

8. Po kliknięciu na `p`, zmień jego czcionkę na pogrubioną i kolor na niebieski.
9. Po naciśnięciu klawisza `Enter` zmień kolor tekstu w `h1` na losowy.
10. Po kliknięciu przycisku zwiększ szerokość `div` o 50px za każdym razem.
11. Po kliknięciu dwukrotnie na `button` zmień jego napis na „Podwójne kliknięcie!”.
12. Po kliknięciu prawym przyciskiem myszy na `p`, wyświetl alert z napisem „Kliknięto prawym przyciskiem”.
13. Stwórz licznik, który zwiększa się o 1 po każdym kliknięciu `button`.
14. Stwórz suwak (`input type="range"`) i wyświetl jego aktualną wartość w `p`.
15. Po najechaniu na `div`, pokaż w `p` współrzędne myszy (X, Y).