

Wyszukiwania elementów w DOM.

document.getElementById("id")

- **Zwraca jeden element** (bo id jest unikalne w dokumencie).
- Używamy, gdy chcemy pobrać konkretny element po jego **identyfikatorze**.

Ćwiczenie:

1. W pliku HTML dodaj:
2. `<h2 id="naglowek">To jest stary tytuł</h2>`
3. W JavaScript pobierz ten element za pomocą `document.getElementById("naglowek")`.
4. Zmień jego tekst na „**Nowy tytuł strony**”.
5. Ustaw mu kolor tekstu na **zielony** i dodaj **podkreślenie**.

Rozwiązanie:

```
<!DOCTYPE html>
```

```
<html lang="pl">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>Ćwiczenie getElementById</title>
```

```
</head>
```

```
<body>
```

```
  <h2 id="naglowek">To jest stary tytuł</h2>
```

```
<script>
```

```
  // Pobranie elementu po ID
```

```
  let naglowek = document.getElementById("naglowek");
```

```
  // Zmiana tekstu
```

```
  naglowek.textContent = "Nowy tytuł strony";
```

```
  // Zmiana koloru tekstu na zielony
```

```
  naglowek.style.color = "green";
```

```
  // Dodanie podkreślenia
```

```
  naglowek.style.textDecoration = "underline";
```

```
</script>
```

```
</body>
```

```
</html>
```

document.getElementsByTagName()

Metoda **document.getElementsByTagName()** służy do **wyszukiwania wszystkich elementów o określonym znaczniku HTML** (np. p, div, h1, img, itp.) w dokumencie.

```
document.getElementsByTagName("nazwa_tagu");
```

przykład:

np. zmiana wyglądu wszystkich paragrafów, nagłówków lub linków.

```
<p>Paragraf 1</p>
```

```
<p>Paragraf 2</p>
```

```
<p>Paragraf 3</p>
```

```
<script>
```

```
let paragrafy = document.getElementsByTagName("p");
```

```
// Ręczne odwołanie się do kolejnych paragrafów
```

```
paragrafy[0].textContent = "Jestem pierwszym paragrafem!";
```

```
paragrafy[1].textContent = "To ja — drugi paragraf!";
```

```
paragrafy[2].textContent = "A ja jestem trzeci!";
```

```
// Dodajmy trochę stylu
```

```
paragrafy[0].style.color = "blue";
```

```
paragrafy[1].style.color = "green";
```

```
paragrafy[2].style.color = "orange";
```

```
</script>
```

```
<p>Paragraf 1</p>
```

```
<p>Paragraf 2</p>
```

```
<p>Paragraf 3</p>
```

```
<script>
```

```
let paragrafy = document.getElementsByTagName("p");
```

```
for (let p of paragrafy) {
```

```
  p.style.color = "blue";
```

```
  p.style.fontWeight = "bold";
```

```
}
```

```
</script>
```

```

```

```

```

```
<script>
```

```
let obrazki = document.getElementsByTagName("img");
```

```
for (let img of obrazki) {
```

```
  img.width = 200; // ustawia szerokość na 200px
```

```
  img.alt = "Zdjęcie zwierzęcia";
```

```
}
```

```
</script>
```

Pętla for...of

Przechodzi po wszystkich elementach kolekcji (np. tablicy, NodeList, HTMLCollection).

```
let owoce = ["jabłko", "banan", "gruszka"];
```

```
for (let owoc of owoce) {
```

```
  console.log(owoc);
```

```
}
```

Tablice:

Tablica (ang. array) to specjalny typ obiektu w JavaScript, który pozwala przechowywać wiele wartości w jednej zmiennej.

Zamiast tworzyć kilka osobnych zmiennych, można umieścić wszystkie dane w jednej strukturze.

```
let owoce = ["jabłko", "banan", "gruszka"];
```

```
let liczby = [1, 2, 3, 4];
```

owoce to nazwa tablicy,

"jabłko", "banan", "gruszka" to jej elementy,

indeksy (czyli pozycje elementów) zaczynają się od 0:

- owoce[0] → "jabłko"
- owoce[1] → "banan"
- owoce[2] → "gruszka"

```
console.log(owoce[1]);
```

Dodawanie i usuwanie elementów

Operacja	Metoda	Opis
Dodaj na koniec	push()	owoce.push("truskawka");
Usuń z końca	pop()	owoce.pop();
Dodaj na początek	unshift()	owoce.unshift("wiśnia");
Usuń z początku	shift()	owoce.shift();

Przechodzenie po elementach tablicy:

```
for (let i = 0; i < owoce.length; i++) {  
  console.log(owoce[i]);  
}  
  
for (let owoc of owoce) {  
  console.log(owoc);  
}
```

Przydatne metody tablic:

Metoda	Działanie	Przykład
length	zwraca długość tablicy	owoce.length
indexOf()	zwraca indeks elementu	owoce.indexOf("banan")
includes()	sprawdza, czy element istnieje	owoce.includes("gruszka")
join()	łączy elementy w string	owoce.join(", ")

Metoda split() w JavaScript dzieli łańcuch znaków na tablicę, rozdzielając go według określonego separatora. Jest to najczęściej stosowana metoda do konwersji tekstu na tablicę w JavaScript.

```
let napis = "jabłko,banan,gruszka";  
let owoce = napis.split(",");  
console.log(owoce); // ["jabłko", "banan", "gruszka"]
```

Ćwiczenie:

Treść zadania

Podpunkt 1 – Utworzenie formularza HTML

1. Stwórz prostą stronę HTML o tytule „Kalkulator ocen”.
2. Umieść pole typu number do wpisywania ocen (zakres 1–6).
3. Umieść przycisk „Dodaj ocenę”, który wywoła funkcję JavaScript po kliknięciu (onclick).
4. Utwórz elementy do wyświetlania wyników:

- Średnia ocen
 - Najwyższa ocena
 - Najniższa ocena
-

Podpunkt 2 – Logika w JavaScript

1. Utwórz tablicę oceny, która przechowuje wszystkie wprowadzone oceny.
 2. Po kliknięciu przycisku:
 - Pobierz wartość z pola input.
 - Sprawdź, czy jest liczbą od 1 do 6.
 - Dodaj ocenę do tablicy.
 - Wywołaj funkcję aktualizujWyniki(), która:
 - Oblicza średnią ocen ręcznie w pętli for.
 - Znajduje najwyższą i najniższą ocenę (można użyć Math.max i Math.min).
 - Aktualizuje odpowiednie elementy HTML z wynikami.
 - Czyści pole formularza i ustawia kursor w polu input.
 3. W przypadku błędnej wartości wyświetl alert z komunikatem „Wprowadź ocenę od 1 do 6!”.
-

Podpunkt 3 – Funkcja aktualizujWyniki()

1. Stwórz funkcję aktualizujWyniki(), która będzie:
 - Ręcznie sumować wszystkie oceny w tablicy oceny.
 - Obliczać średnią i wyświetlać ją w elemencie sredniaOceny.
 - Znajdować najwyższą ocenę i wyświetlać w elemencie najwyzszaOcena.
 - Znajdować najniższą ocenę i wyświetlać w elemencie najnizszaOcena.
-

Podpunkt 4 – Dodatkowe wymagania

- Pole formularza powinno być czyszczone po dodaniu oceny.
- Kursor powinien wracać do pola input po dodaniu oceny.
- Obsługa błędnych danych (ocena poza zakresem 1–6).

Kalkulator ocen

Dodaj ocenę (1-6):

Dodaj ocenę

Średnia ocen: 0

Najwyższa ocena: 0

Najniższa ocena: 0

Kalkulator ocen

Dodaj ocenę (1-6):

Dodaj ocenę

Średnia ocen: 4.67

Najwyższa ocena: 5

Najniższa ocena: 4

Podpowiedzi:

let oceny = [];

- Funkcje wbudowane:

Math.max(...oceny); // najwyższa ocena

Math.min(...oceny); // najniższa ocena

const ocena = Number(document.getElementById('ocena').value);

Number() – konwertuje wartość z tekstu na liczbę.

sredniaOcen.textContent = srednia.toFixed(2); // 2 miejsca po przecinku

ocenaInput.focus();

focus() to wbudowana metoda w JavaScript dla elementów formularza (input, textarea, select).

Służy do **ustawienia kursora w danym polu**, czyli sprawia, że użytkownik może od razu pisać w tym polu bez klikania.

```
ocenalInput.value = "";    // Czyści pole formularza
```

```
ocenalInput.focus();      // Ustawia kursor w polu
```

Zadanie domowe:

https://ee-informatyk.pl/inf03-ee09/praktyka/INF.03-02-22.06-SG?utm_source=chatgpt.com

