**1. Якими способами можна підключити JavaScript до HTML?**

Є кілька способів, залежно від того, наскільки зручно тобі працювати:

1. **Прямо в HTML через тег <script>**  
   Просто пишеш код всередині HTML:
2. <script>
3. console.log("Привіт із JS!");
4. </script>
5. **Підключаєш окремий JS-файл**  
   Це зручніше, якщо коду багато:
6. <script src="script.js"></script>
7. **Через defer або async**  
   Щоб не зламати завантаження сторінки:
   * **defer** — скрипт запуститься після завантаження HTML.
   * **async** — запускається відразу після завантаження скрипта, може бути хаос, якщо кілька скриптів.
8. <script src="script.js" defer></script>

**2. Як отримати доступ до елементів на сторінці (DOM) через JS?**

Уяви, що DOM — це як велика карта всіх елементів на сторінці, і тобі треба знайти потрібний.

* **getElementById('id')** → шукає елемент за **ID**
* const title = document.getElementById('main-title');
* **getElementsByClassName('class')** → шукає всі елементи з таким класом
* const items = document.getElementsByClassName('list-item');
* **getElementsByTagName('tag')** → шукає всі елементи з певним тегом
* const paragraphs = document.getElementsByTagName('p');
* **querySelector('selector')** → повертає перший елемент, що підходить
* const firstButton = document.querySelector('button');
* **querySelectorAll('selector')** → повертає ВСІ підходящі елементи
* const allButtons = document.querySelectorAll('button');

**3. Чим відрізняється querySelector від querySelectorAll?**

Думай про це так:

* **querySelector** — бере тільки **перший** знайдений елемент.
* const firstH1 = document.querySelector('h1'); // тільки перший <h1>
* **querySelectorAll** — бере **всі** елементи, які підходять під селектор.
* const allH1s = document.querySelectorAll('h1'); // всі <h1> на сторінці

**Фішка:**

* **querySelector → одиночний елемент**
* **querySelectorAll → список (NodeList)**, з яким можна працювати через цикл.

**4. Які методи повертають колекцію елементів?**

Якщо тобі треба **кілька елементів**, використовуй ці методи:

| **Метод** | **Тип колекції** | **Жива чи статична?** |
| --- | --- | --- |
| getElementsByClassName() | HTMLCollection | Жива |
| getElementsByTagName() | HTMLCollection | Жива |
| querySelectorAll() | NodeList | Статична |

* **HTMLCollection** — змінюється, якщо щось змінюєш на сторінці.
* **NodeList** — залишиться таким, як отримав, навіть якщо DOM зміниться.

**5. Що таке івенти HTML і як їх використовувати?**

**Івенти** — це просто реакції на дії користувача. Наприклад:

* **click** — коли клікаєш на кнопку
* **mouseover** — коли наводиш мишку
* **keydown** — коли натискаєш клавішу
* **submit** — коли відправляєш форму

**Приклад:**

<button id="btn">Натисни мене</button>

<script>

const btn = document.getElementById('btn');

btn.addEventListener('click', () => {

alert('Ти натиснув кнопку!');

});

</script>

**6. Що таке функція у JS і її аргументи?**

**Функція** — це просто набір інструкцій, який ти можеш викликати багато разів.

**Приклад простої функції:**

function greet(name) {

console.log('Привіт, ' + name + '!');

}

greet('Анна'); // Виведе: Привіт, Анна!

**Аргументи** — це дані, які ти передаєш функції при виклику ('Анна' у прикладі).

**Інші способи написання функцій:**

1. **Анонімна функція:**
2. const sayHi = function() {
3. console.log('Привіт!');
4. };
5. **Стрілкова функція (Arrow Function):**
6. const sum = (a, b) => a + b;
7. console.log(sum(3, 4)); // 7