**Lab 9: Một số chủ đề JavaScript (2)**

**1. Arrow Function (Hàm mũi tên)**

* **Cú pháp**
  + Không tham số: () => …
  + Một tham số: x => …
  + Nhiều tham số: (x, y) => …
* **Ưu điểm**
  + Viết ngắn gọn, rõ ràng
  + Tự động return nếu thân hàm chỉ một biểu thức
* **Lưu ý về this**
  + Kế thừa this từ phạm vi bên ngoài, tránh sai ngữ cảnh trong callback
* **Ví dụ**
* const sq = x => x \* x;
* console.log(sq(5)); // 25
* setTimeout(() => console.log("Đã xong 4s"), 4000);

**2. Asynchronous Programming (Bất đồng bộ)**

* **Vấn đề đồng bộ**
  + Một tác vụ “nặng” sẽ chặn cả chương trình (ví dụ vòng while chờ 5 s).
* **Cách giải quyết**
  + **Callback**: truyền hàm vào, gọi khi xong → dễ rối nếu lồng sâu.
  + **Promise**: .then()/.catch() rõ ràng hơn.
  + **async/await**: viết bất đồng bộ như đồng bộ, dễ đọc nhất.
* **Ví dụ Promise + async/await**
* fetch('/api/data')
* .then(res => res.json())
* .then(data => console.log(data))
* .catch(err => console.error(err));
* // Với async/await
* async function load() {
* try {
* const res = await fetch('/api/data');
* const data = await res.json();
* console.log(data);
* } catch(e) {
* console.error(e);
* }
* }
* load();

**3. Bài tập nhanh**

* **Cài đặt** các ví dụ arrow function và async/await.
* **Trắc nghiệm**
  + B Arrow function **không** có ngữ cảnh this riêng.
  + C Callback lồng nhau **không** giúp code dễ đọc hơn.