

OBJECT TRONG JAVASCRIPT

Tổng quan

- Khái niệm
 - Object là kiểu dữ liệu tổng hợp
 - Dùng để lưu trữ dữ liệu có cấu trúc
 - Tập hợp các cặp khóa-giá trị
- Cặp khóa và giá trị
 - Khóa (key): Chuỗi hoặc Symbol
 - Giá trị (value): Bất kỳ kiểu dữ liệu nào
- Thuộc tính (Properties)
 - Cặp khóa-giá trị lưu trữ dữ liệu
 - Ví dụ: ten: "Nguyen Van A"
- Phương thức (Methods)
 - Thuộc tính có giá trị là hàm
 - Thực hiện hành động
 - Ví dụ: noiChao: function() { console.log("Xin chào"); }
- So sánh với mảng (Array)
 - Object: Dữ liệu không có thứ tự, truy cập bằng khóa
 - Array: Dữ liệu có thứ tự, truy cập bằng chỉ số (index)
- So sánh với kiểu dữ liệu nguyên thủy (Primitive Types)
 - Object: Kiểu tham chiếu (reference type)
 - Primitive: Kiểu giá trị (value type) (String, Number, Boolean, Null, Undefined, Symbol, BigInt)
- Vai trò trong lập trình
 - Đại diện cho các thực thể trong thế giới thực
 - Tổ chức dữ liệu phức tạp
 - Xây dựng cấu trúc dữ liệu tùy chỉnh

Truy cập thuộc tính

- Dấu chấm (.)
 - Cách truy cập phổ biến
 - Dùng khi biết tên thuộc tính
 - Ví dụ: sinhVien.ten
- Ngoặc vuông []
 - Dùng khi tên thuộc tính là biến
 - Dùng khi tên thuộc tính chứa ký tự đặc biệt (khoảng trắng, dấu gạch ngang)
 - Ví dụ: sinhVien["tuoi"]
- Gọi phương thức
 - Sử dụng dấu chấm hoặc ngoặc vuông
 - Thêm () sau tên phương thức
 - Ví dụ: sinhVien.noiChao()
- So sánh hai cách truy cập
 - Dấu chấm: Ngắn gọn, dễ đọc, tên thuộc tính cố định
 - Ngoặc vuông: Linh hoạt, truy cập động, tên thuộc tính có thể là biểu thức
- Tên thuộc tính đặc biệt
 - Chứa khoảng trắng, dấu gạch ngang
 - Bắt buộc dùng ngoặc vuông
 - Ví dụ: doiTuong["ten-san-pham"]
- Tên thuộc tính nằm trong biến
 - Sử dụng ngoặc vuông với biến chứa tên thuộc tính
 - Ví dụ: let key = "diaChi"; doiTuong[key]

Duyệt đối tượng

- Vòng lặp for...in
 - Lặp qua các khóa (key) của đối tượng
 - Ví dụ: javascript for (let key in doiTuong) { // key là tên thuộc tính // doiTuong[key] là giá trị thuộc tính }
- Biến đại diện cho khóa
 - Biến key trong for...in
 - Chứa tên thuộc tính dưới dạng chuỗi
- Truy cập động bằng biến
 - Sử dụng biến key với ngoặc vuông để lấy giá trị
 - Ví dụ: doiTuong[key]
- Lấy danh sách khóa
 - Object.keys(doiTuong)
 - Trả về mảng các chuỗi tên thuộc tính
- Lấy danh sách giá trị
 - Object.values(doiTuong)
 - Trả về mảng các giá trị thuộc tính
- Lấy cặp khóa và giá trị
 - Object.entries(doiTuong)
 - Trả về mảng các mảng [key, value]

Ứng dụng CRUD (Create, Read, Update, Delete)

- Thêm dữ liệu (Create)
 - Tạo đối tượng mới
 - Thêm đối tượng vào mảng đối tượng
 - Ví dụ: Thêm sinh viên mới vào danhSachSinhVien
- Hiển thị dữ liệu (Read)
 - Duyệt mảng đối tượng
 - In ra các thuộc tính của từng đối tượng
 - Ví dụ: Hiển thị thông tin tất cả sinh viên
- Cập nhật dữ liệu (Update)
 - Tìm đối tượng cần cập nhật (theo ID hoặc thuộc tính khác)
 - Thay đổi giá trị thuộc tính của đối tượng đó
 - Ví dụ: Cập nhật điểm của một sinh viên
- Xóa dữ liệu (Delete)
 - Tìm đối tượng cần xóa
 - Xóa đối tượng khỏi mảng (sử dụng splice hoặc filter)
 - Ví dụ: Xóa một sinh viên khỏi danh sách
- Kiểm tra dữ liệu
 - Tìm kiếm đối tượng theo điều kiện
 - Kiểm tra sự tồn tại của thuộc tính
 - Ví dụ: Kiểm tra sinh viên có tên "An" hay không
- Tối ưu tìm kiếm
 - Sắp xếp mảng đối tượng
 - Sử dụng các phương thức tìm kiếm hiệu quả (find, findIndex)

Mảng đối tượng (Array of Objects)

- Khái niệm
 - Mảng mà mỗi phần tử là một đối tượng
 - Dùng để lưu trữ danh sách các thực thể có cùng cấu trúc
 - Ví dụ: Danh sách sinh viên, danh sách sản phẩm
- Truy cập theo vị trí
 - Sử dụng chỉ số mảng
 - Ví dụ: danhSachSinhVien[0]
- Truy cập thuộc tính
 - Sau khi truy cập đối tượng, dùng dấu chấm hoặc ngoặc vuông để truy cập thuộc tính
 - Ví dụ: danhSachSinhVien[0].ten
- Duyệt danh sách
 - Sử dụng for loop, forEach, map, filter, v.v.
 - Ví dụ: javascript danhSachSinhVien.forEach(sv => { console.log(sv.ten); });
- So sánh với mảng đơn
 - Mảng đơn: Chứa các giá trị nguyên thủy hoặc đối tượng không có cấu trúc phức tạp
 - Mảng đối tượng: Mỗi phần tử là một đối tượng có cấu trúc rõ ràng, chứa nhiều thuộc tính
- So sánh với một đối tượng
 - Một đối tượng: Đại diện cho một thực thể duy nhất
 - Mảng đối tượng: Đại diện cho một tập hợp các thực thể tương tự

Thêm sửa xóa

- Thêm thuộc tính
 - Gán giá trị cho thuộc tính mới
 - Ví dụ: doiTuong.email = "a@example.com"
- Cập nhật giá trị
 - Gán giá trị mới cho thuộc tính hiện có
 - Ví dụ: doiTuong.tuoi = 30
- Xóa thuộc tính
 - Dùng toán tử delete
 - Ví dụ: delete doiTuong.diaChi
- Thêm phương thức
 - Gán hàm cho thuộc tính mới
 - Ví dụ: doiTuong.tinhTong = function(a, b) { return a + b; }
- Phân biệt thêm và cập nhật
 - Nếu thuộc tính chưa tồn tại: Thêm mới
 - Nếu thuộc tính đã tồn tại: Cập nhật giá trị