

NỘI DUNG ÔN TẬP

CẤU TRÚC DỮ LIỆU (DATA STRUCTURES) VÀ TOÁN TỬ HIỆN ĐẠI (MODERN OPERATORS)

Nội dung chính:

Cấu trúc dữ liệu, hủy cấu trúc, toán tử trải rộng (spread) (...), toán tử đoản mạch (short circuiting) (&& ||), toán tử vô hiệu (nullish) (??), chuỗi tùy chọn (optional chaining),...

Bài tập minh họa:

1. Tạo mảng:

```
arr = ["t", "c", "e", "j", "b", "o", "-", "y", "a", "r", "r", "a"]
```

- 1.1. Xóa phần tử "-" khỏi mảng thêm vào " " thay thế
- 1.2. In ra màn hình chuỗi in hoa đảo ngược của mảng
- 1.3. Tìm và thay thế chuỗi "object" thành chuỗi "strings"
- 1.4. Tạo chuỗi "data structures", sử dụng concat nối 2 chuỗi
- 1.5. Cắt và in ra màn hình "structures" từ chuỗi được nối ở trên

2. Cho 2 mảng sau

```
arr1 = [14, 12, 23, 43, 35, 30]
arr2 = [91, 82, 19, 73, 46, 20]
```

- 2.1. Tạo mảng arr bao gồm arr1 và arr2 biết arr2 nằm ở vị trí thứ 2 trong arr1 (sử dụng toán tử spread)
- 2.2. In ra màn hình các số chẵn nhỏ hơn 20 hoặc lớn hơn 70 (sử dụng toán tử đoản mạch)
- 2.3. In ra màn hình các số nguyên tố trong khoảng từ 30 đến 80 (sử dụng toán tử đoản mạch)
- 2.4. In ra mảng số lẻ được sắp xếp giảm dần
3. Tạo form đăng nhập, sử dụng biểu thức chính quy để kiểm tra
 - 3.1. Tên đăng nhập bắt đầu bằng "PS" tiếp theo là 5 ký tự số
 - 3.2. Mật khẩu có ít nhất 7 ký tự bao gồm chữ in hoa, số và ký tự đặc biệt
4. Cho đối tượng sau:

```
let course = {
  name: 'js nâng cao',
  id: 'web2062',
}
```

Sử dụng toán tử optional chaining (?) và toán tử nullish (??) để in ra giá trị của thuộc tính lesson: 'Modern Operators'

HÀM (FUNCTIONS)

Nội dung chính

Hàm, hàm ẩn danh (anonymous function), hàm mũi tên (arrow function), hàm gọi lại (callback function); hàm trả có trả về hàm; tham số hàm, tham số mặc định,...

Bài tập minh họa

```
function sortArray(arr) {  
    arr.sort((a,b) => a-b)  
    return arr  
}
```

1. Cho hàm sắp xếp tăng dần như mẫu
 - 1.1. Viết lại hàm sortArray theo dạng around function
 - 1.2. Viết lại đối số của hàm sort theo dạng anonymous function
 - 1.3. Nhập mảng sử dụng hàm callback hiển thị mảng được sắp xếp tăng dần
2. Cho người dùng nhập vào mảng số tự nhiên (mảng có ít nhất 10 số), viết các hàm sau dưới dạng around function hoặc callback function
 - 2.1. Viết hàm tìm số lớn nhất trong mảng, in ra màn hình số lớn nhất
 - 2.2. Viết hàm đảo ngược thứ tự các phần tử, in ra màn hình mảng đảo ngược
 - 2.3. Cho người dùng nhập vào 1 số bất kỳ, viết hàm kiểm tra xem số đó có nằm trong mảng hay không
 - 2.4. Viết hàm xóa một phần tử khỏi mảng, in ra màn hình mảng mới
 - 2.5. Viết hàm sắp xếp phần tử số lẻ theo thứ tự giảm dần

MẢNG (ARRAYS) VÀ LÀM VIỆC VỚI MẢNG

Nội dung chính

Mảng, cách khởi tạo và các phương thức xử lý mảng (duyệt, cắt, ghép, tìm kiếm, sắp xếp, làm phẳng,...).

Bài tập minh họa

1. Tạo mảng $A = [1, 3, 5, 7, 9, \dots, n]$ với n do người dùng nhập (prompt) mà không dùng vòng lặp: for, forEach hoặc while.
2. Mảng B được tạo bằng cách gấp đôi giá trị mỗi phần tử của mảng A.
3. Mảng C được tạo bằng cách sắp xếp mảng B theo thứ tự bảng chữ cái từ Z-A.
4. Mảng D được tạo bằng cách lấy từ mảng A các phần tử chia hết cho 7.
5. Mảng E được tạo bằng cách ghép mảng A vào cuối mảng B.
6. Cho biết mảng E có chứa 1 số nào đó chia hết cho 5 mà lớn hơn 50 hay không.

DOM NÂNG CAO (ADVANCED DOM) VÀ SỰ KIỆN (EVENTS)

Nội dung chính

Truy xuất DOM; thuộc tính và giá trị của DOM; các cách thêm, xóa các sự kiện; các sự kiện thường dùng; tham số sự kiện.

Bài tập minh họa

1. Khi bấm vào “Bốc số” sẽ tạo ra 1 số ngẫu nhiên trong đoạn [1,90] không trùng với những số trước đó. Số vừa “bốc” được sẽ lưu vào mảng và hiển thị (log) mảng đó ra.
2. Nút “Lịch sử” sẽ hiển thị tất cả các số trong mảng đã lưu ra màn hình thành các thẻ button. Khi bấm vào button số nào thì button đó đổi màu và không cho phép bấm nữa.
3. Ô (input) “dò số” cho phép người dùng nhập vào các số bất kỳ cách nhau khoảng trắng (space). Khi bấm nút “Kiểm tra” sẽ kiểm tra các số đã nhập có hoàn toàn trùng với các số đã bốc. Thông báo (alert) “Trùng (n) số” với n là số lượng trùng.
4. Nút “Chơi lại” sẽ đặt lại các giá trị mặc định như khi chưa bắt đầu chơi (Không được tải lại/reload trang).
5. Khi trò chơi đang diễn ra mà bấm tải lại/reload trang thì phải hiện thông báo xác nhận (prompt): “Trò chơi đang diễn ra! Bạn có chắc muốn tải lại trang không?”. Khi được xác nhận “Yes” thì mới tải lại trang, ngược lại không tải lại trang.

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (OOP) TRONG JAVASCRIPT

Nội dung chính

Tạo đối tượng, tạo lớp đối tượng; các thuộc tính và phương thức trong đối tượng; các tính chất của OOP: đóng gói, trừu tượng, kế thừa, đa hình;

Bài tập minh họa

1. Xây dựng lớp đối tượng **HocSinh** với các thuộc tính: HoTen, NgaySinh (dd/mm/yyyy), Lop, Diem và các phương thức: tinhtTuoi() bằng cách lấy ngày, tháng, năm hiện tại trừ cho NgaySinh và làm tròn xuống; xepLoai() dựa theo Diem và tra trong bảng sau:

Điểm	[0,3)	[3,6)	(6,8]	(8,10]
Xếp loại	Yếu	Trung bình	Khá	Giỏi

2. Xây dựng lớp đối tượng **SinhVien** bằng cách kế thừa các thuộc tính và phương thức từ lớp **HocSinh**. Bổ sung phương thức thiBaoVe(monThi) và đưa ra thông báo (log) “Sinh viên HoTen, lớp Lop đủ điều kiện dự thi monThi” hoặc “Sinh viên HoTen, lớp Lop bị cấm thi môn monThi do điểm quá trình dưới 5”.
3. Xây dựng lớp đối tượng **NhanVien** với các thuộc tính: HoTen, NgayNhanViec (dd/mm/yyyy), HeSoLuong và các phương thức: tinhtLuong(luongCoBan, soNgayLamViec) trả về lương thực nhận = luongCoBan x HeSoLuong x soNgayLamViec / 26; tinhtThuongTet(soThangLamViec) trả về thưởng Tết = luongCoBan x 26 x soThangLamViec / 12.

4. Xây dựng lớp đối tượng **CanBo** bằng cách kế thừa các thuộc tính và phương thức từ lớp **NhanVien**. Bổ sung thuộc tính phuCapChucVu và viết lại phương thức tinhLuong(luongCoBan) trả về lương thực nhận = luongCoBan x (HeSoLuong + phuCapChucVu) x soNgayLamViec / 26.
5. Xây dựng lớp đối tượng **SanPham** với các thuộc tính: TenSanPham, GiaBan, HanSuDung (dd/mm/yyyy) và các phương thức kiemTraHSD() trả về true hoặc false dựa vào HanSuDung so với thời điểm hiện tại; phương thức tieuHuy() trả về thông báo (log) "Sản phẩm TenSanPham đã bị tiêu hủy vì quá hạn sử dụng ? ngày"; phương thức hủy thông báo (log) "Đã tiêu hủy thành công!" mỗi khi có sản phẩm bị tiêu hủy.
6. Tạo mới các đối tượng để minh họa cho các lớp đối tượng bên trên; gọi các phương thức và kiểm tra kết quả.

BẮT ĐỒNG BỘ (ASYNCHRONOUS) VÀ AJAX

Nội dung chính

Xử lý bất đồng bộ bằng callback function, promise, async/await; gửi nhận dữ liệu bằng Fetch API

Bài tập minh họa

Cho link sau:

<https://raw.githubusercontent.com/kenzouno1/DiaGioiHanhChinhVN/master/data.json>

1. Xây dựng hàm loadTinh() đọc dữ liệu và hiển thị danh sách tất cả 63 tỉnh/thành phố trực thuộc trung ương ở Việt Nam.
2. Xây dựng hàm loadHuyen(idTinh) đọc dữ liệu và hiển thị các quận/huyện của thành phố theo idTinh.
3. Xây dựng hàm loadPhuongXa(idHuyen, loai="Phường") đọc dữ liệu và hiển thị các loai ("Phường" hoặc "Xã") tương ứng theo idHuyen.

Cho link sau:

(A) <https://fakestoreapi.com/products/categories>

(B) [https://fakestoreapi.com/products/category/\[tenCatagory\]](https://fakestoreapi.com/products/category/[tenCatagory])

(C) [https://fakestoreapi.com/products/\[maSP\]](https://fakestoreapi.com/products/[maSP])

4. Thực hiện lấy 2 danh mục ngẫu nhiên từ link (A). Sau đó, lấy hết các sản phẩm có trong 2 danh mục vừa chọn và hiển thị tên (title) kèm giá (price) và nút xóa.
5. Sử dụng link (C) và hiển thực nút xóa ở câu 4. Hiển thị thông báo (alert) nếu xóa thành công!