**Q1: Can you create a simple HTML search input with a button to submit a keyword?**



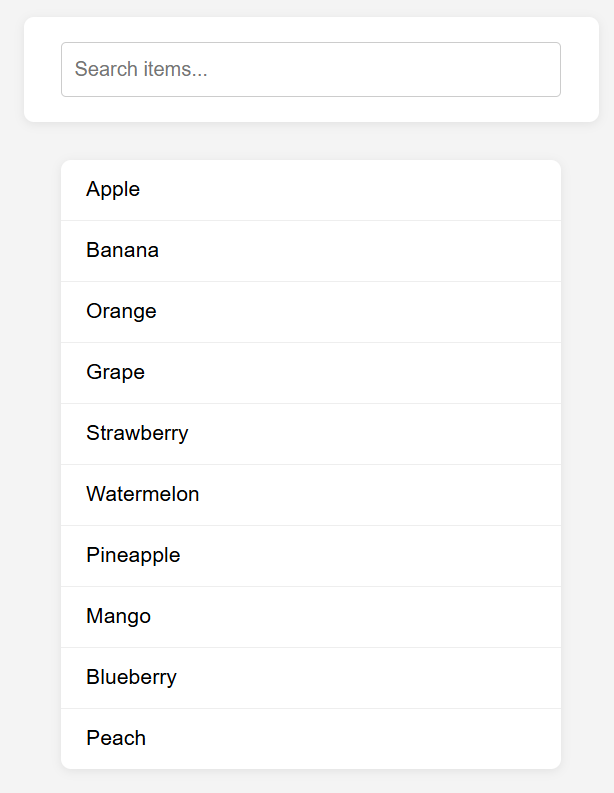
<form action="#" method="get">

<input type="text" name="keyword" placeholder="Enter keyword..." />

<button type="submit">Search</button>

</form>

**Q2: Build an HTML search bar styled with CSS that filters a list of items in real time using JavaScript as the user types.**

****

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Real-Time Search Filter</title>

<style>

body {

font-family: Arial, sans-serif;

background: #f4f4f4;

margin: 0;

padding: 40px;

}

.search-container {

max-width: 400px;

margin: 0 auto 30px auto;

background: #fff;

padding: 20px 30px;

border-radius: 8px;

box-shadow: 0 2px 8px rgba(0,0,0,0.08);

}

.search-container input[type="text"] {

width: 100%;

padding: 12px 10px;

border: 1px solid #ccc;

border-radius: 4px;

font-size: 16px;

box-sizing: border-box;

}

ul.item-list {

max-width: 400px;

margin: 0 auto;

padding: 0;

list-style: none;

background: #fff;

border-radius: 8px;

box-shadow: 0 2px 8px rgba(0,0,0,0.08);

}

ul.item-list li {

padding: 14px 20px;

border-bottom: 1px solid #eee;

font-size: 17px;

transition: background 0.2s;

}

ul.item-list li:last-child {

border-bottom: none;

}

ul.item-list li.hide {

display: none;

}

ul.item-list li:hover {

background: #f0f8ff;

}

</style>

</head>

<body>

<div class="search-container">

<input type="text" id="searchBar" placeholder="Search items...">

</div>

<ul class="item-list" id="itemList">

<li>Apple</li>

<li>Banana</li>

<li>Orange</li>

<li>Grape</li>

<li>Strawberry</li>

<li>Watermelon</li>

<li>Pineapple</li>

<li>Mango</li>

<li>Blueberry</li>

<li>Peach</li>

</ul>

<script>

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

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

const items = itemList.getElementsByTagName('li');

searchBar.addEventListener('input', function() {

const filter = searchBar.value.toLowerCase();

for (let i = 0; i < items.length; i++) {

const text = items[i].textContent.toLowerCase();

if (text.includes(filter)) {

items[i].classList.remove('hide');

} else {

items[i].classList.add('hide');

}

}

});

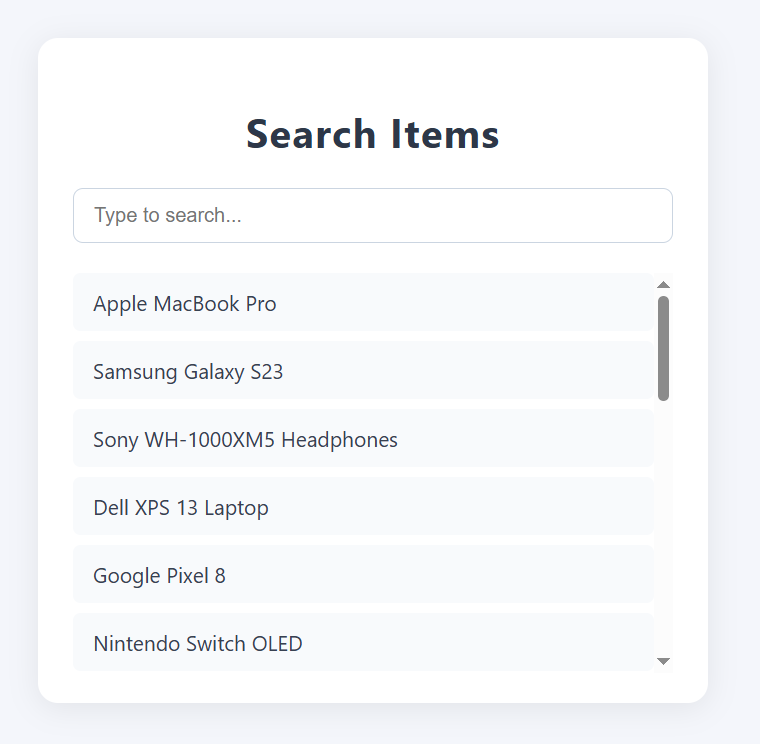
</script>

</body>

</html>

**Q3: Create a fully styled HTML, CSS, and JavaScript search interface that allows users to filter a dynamic list of items using keywords and highlights the matching results.**

**Información adicional: In one file.**



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Dynamic Search Interface</title>

<style>

body {

font-family: 'Segoe UI', Arial, sans-serif;

background: #f4f6fb;

margin: 0;

padding: 0;

display: flex;

flex-direction: column;

align-items: center;

min-height: 100vh;

}

.container {

background: #fff;

margin-top: 60px;

padding: 32px 28px 24px 28px;

border-radius: 16px;

box-shadow: 0 4px 24px rgba(0,0,0,0.08);

width: 100%;

max-width: 480px;

}

h1 {

text-align: center;

color: #2d3748;

margin-bottom: 24px;

font-size: 2rem;

letter-spacing: 1px;

}

.search-box {

width: 100%;

padding: 12px 16px;

border: 1.5px solid #cbd5e1;

border-radius: 8px;

font-size: 1rem;

margin-bottom: 24px;

transition: border 0.2s;

outline: none;

}

.search-box:focus {

border-color: #6366f1;

box-shadow: 0 0 0 2px #6366f133;

}

ul.item-list {

list-style: none;

padding: 0;

margin: 0;

max-height: 320px;

overflow-y: auto;

}

ul.item-list li {

padding: 12px 16px;

border-radius: 6px;

margin-bottom: 8px;

background: #f8fafc;

color: #374151;

font-size: 1.05rem;

transition: background 0.2s;

position: relative;

}

ul.item-list li:last-child {

margin-bottom: 0;

}

ul.item-list li.highlighted {

background: #e0e7ff;

color: #3730a3;

}

.highlight {

background: #fde68a;

color: #b45309;

border-radius: 3px;

padding: 0 2px;

}

.no-results {

text-align: center;

color: #a0aec0;

margin-top: 16px;

font-size: 1.1rem;

}

@media (max-width: 600px) {

.container {

max-width: 98vw;

padding: 18px 6px 12px 6px;

}

h1 {

font-size: 1.3rem;

}

}

</style>

</head>

<body>

<div class="container">

<h1>Search Items</h1>

<input type="text" id="searchInput" class="search-box" placeholder="Type to search..." autocomplete="off" />

<ul id="itemList" class="item-list"></ul>

<div id="noResults" class="no-results" style="display:none;">No results found.</div>

</div>

<script>

// Example dynamic items (can be replaced with any data source)

const items = [

'Apple MacBook Pro',

'Samsung Galaxy S23',

'Sony WH-1000XM5 Headphones',

'Dell XPS 13 Laptop',

'Google Pixel 8',

'Nintendo Switch OLED',

'Canon EOS R6 Camera',

'Bose QuietComfort Earbuds',

'Microsoft Surface Pro 9',

'Amazon Kindle Paperwhite',

'Logitech MX Master 3S Mouse',

'Apple iPad Air',

'Fitbit Versa 4',

'HP Envy 15',

'JBL Flip 6 Speaker',

'Razer DeathAdder V2',

'Asus ROG Zephyrus G14',

'Lenovo ThinkPad X1 Carbon',

'GoPro HERO11',

'DJI Mini 3 Pro Drone'

];

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

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

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

function highlightMatch(text, keyword) {

if (!keyword) return text;

// Escape regex special characters in keyword

const safeKeyword = keyword.replace(/[.\*+?^${}()|[\]\\]/g, '\\$&');

const regex = new RegExp(`(${safeKeyword})`, 'gi');

return text.replace(regex, '<span class="highlight">$1</span>');

}

function renderList(filter = '') {

itemList.innerHTML = '';

let found = false;

const trimmedFilter = filter.trim();

items.forEach(item => {

if (!trimmedFilter || item.toLowerCase().includes(trimmedFilter.toLowerCase())) {

found = true;

const li = document.createElement('li');

li.innerHTML = highlightMatch(item, trimmedFilter);

if (trimmedFilter) li.classList.add('highlighted');

itemList.appendChild(li);

}

});

noResults.style.display = found ? 'none' : 'block';

}

searchInput.addEventListener('input', (e) => {

renderList(e.target.value);

});

// Initial render

renderList();

</script>

</body>

</html>