

## Topic 1: Advanced DOM Selection with querySelectorAll

querySelectorAll اختيار مجموعة من العناصر باستخدام

```
// HTML :
// <div class="product-card">
//   <h2 class="product-title">Product 1</h2>
//   <span class="price">$99</span>
// </div>
// <div class="product-card">
//   <h2 class="product-title">Product 2</h2>
//   <span class="price">$149</span>
// </div>

// Select all product cards and update their style
const products = document.querySelectorAll('.product-card');
products.forEach(product => {
  product.style.border = '1px solid #ccc';
  const price = product.querySelector('.price');
  if (parseInt(price.textContent.slice(1)) > 100) {
    product.classList.add('premium');
  }
});
```

ويعيد قائمة يمكن التكرار عليها (selector) يختار كل العناصر التي تطابق المحدد querySelectorAll.

## Topic 2: Creating and Inserting Elements

إنشاء عناصر جديدة وإضافتها للصفحة

```
// Real-life example: Adding a notification message
function showNotification(message, type = 'info') {
  // Create the notification element
  const notification = document.createElement('div');
  notification.className = `notification ${type}`;
  notification.textContent = message;

  // Create close button
  const closeBtn = document.createElement('button');
  closeBtn.textContent = 'x';
  closeBtn.className = 'close-btn';
  notification.appendChild(closeBtn);

  // Add to page
  document.body.insertBefore(notification, document.body.firstChild);

  // Remove after 5 seconds
  setTimeout(() => notification.remove(), 5000);
}
```

```
// Close button click handler
closeBtn.onclick = () => notification.remove();
}

// Usage
showNotification('Your changes have been saved!', 'success');
showNotification('Please check your input', 'error');
```

createElement و appendChild مثال عملي لإنشاء نظام إشعارات ديناميكي باستخدام

## Bonus Practical Exercise

تمرين عملي: قم بإنشاء نموذج تسجيل مع التحقق من صحة المدخلات مباشرة وإظهار رسائل الخطأ بشكل ديناميكي

```
// Exercise skeleton code
function createRegistrationForm() {
  const form = document.createElement('form');
  form.className = 'registration-form';

  // Add form fields
  const fields = [
    { type: 'text', name: 'username', validate: 'required' },
    { type: 'email', name: 'email', validate: 'email' },
    { type: 'tel', name: 'phone', validate: 'phone' }
  ];

  fields.forEach(field => {
    // Create and append form fields
    // Add validation listeners
    // Show/hide error messages
  });

  return form;
}
```

## Key Concepts Covered

المفاهيم الرئيسية التي تم تغطيتها في هذه الجلسة:

- 1. **querySelectorAll**: Selecting multiple elements at once
- 2. **Dynamic Element Creation**: Creating and inserting new DOM elements

## Advanced: Form Validation and Dynamic UI

التحقق من صحة النماذج وتحديث واجهة المستخدم بشكل ديناميكي

Feature	Example Usage	Common Use Case
classList	.add/.remove	State management
createElement	Dynamic content	Notifications

Feature	Example Usage	Common Use Case
querySelectorAll	Bulk updates	List operations

استخدام هذه الميزات يساعد في إنشاء واجهات مستخدم تفاعلية وديناميكية