FORMATION
HTML, CSS ET JAVASCRIPT

INTÉGRATION HTML-CSS-JS

Créer des pages web interactives complètes

📚 PLAN DE FORMATION

PARTIE 1 : Les Bases de l'Intégration

- Comment structurer un projet web
- Lier HTML, CSS et JavaScript
- Organisation des fichiers
- Exercice : Structure de base

PARTIE 2 : Composants Interactifs

- Bouton avec effet au clic (HTML + CSS + JS)
- Menu déroulant interactif
- Modal/Popup
- Carrousel d'images
- 5 exercices corrigés

PARTIE 3 : Formulaires Dynamiques

- Validation en temps réel
- Messages d'erreur stylés
- Formulaire de contact complet
- 3 exercices corrigés

PARTIE 4 : Applications Complètes

- Calculatrice interactive
- Quiz interactif
- Todo List avancée
- Météo App (avec API)
- 4 exercices corrigés

🎯 12 exercices pratiques avec corrections détaillées

PARTIE 1 — Les Bases de l'Intégration

1.1 Structure d'un projet web

Voici la structure recommandée pour organiser vos fichiers :

📄 **structure-projet.txt**

```
📁 mon-projet/
  📄 index.html        ← Page principale
  📄 style.css         ← Styles
  📄 script.js         ← JavaScript
  📁 images/           ← Images du projet
    🖼️ logo.png
    🖼️ banniere.jpg
  📁 css/              ← CSS supplémentaires (optionnel)
  📁 js/               ← JS supplémentaires (optionnel)
```

1.2 Lier les fichiers ensemble

📄 **index.html**

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Mon Projet Web</title>
```

```
  <!-- Lien vers le CSS -->
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <h1 id="titre">Bienvenue sur mon site</h1>
  <button id="monBouton">Cliquez-moi !</button>

  <!-- Lien vers JavaScript (à la fin du body) -->
  <script src="script.js"></script>
</body>
</html>
```

📄 **style.css**

```
/* Styles globaux */
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 20px;
  background: #f5f5f5;
}

#titre {
  color: #3B82F6;
  text-align: center;
}
```

```
#monBouton {
  background: #10B981;
  color: white;
  border: none;
  padding: 15px 30px;
  border-radius: 8px;
  cursor: pointer;
  font-size: 16px;
  display: block;
  margin: 20px auto;
}

#monBouton:hover {
  background: #059669;
}
```

📄 **script.js**

```
// Sélectionner les éléments
const titre = document.getElementById('titre');
const bouton = document.getElementById('monBouton');

// Ajouter un événement au clic
bouton.addEventListener('click', () => {
  titre.textContent = 'Vous avez cliqué !';
  titre.style.color = '#10B981';
});
```

> ⚠️ **Important**
>
> Mettez toujours la balise <script> à la fin du <body> pour que le DOM soit chargé avant l'exécution du JavaScript.

> ✏️ **EXERCICE 1 : Votre premier projet intégré**
>
> Créez un projet avec la structure ci-dessus. Ajoutez un paragraphe <p> et un bouton qui change le texte du paragraphe au clic. Stylisez avec du CSS (couleurs, espacements).

CORRECTION EXERCICE 1 :

📄 **index.html**

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Exercice 1</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <h1>Mon Premier Projet</h1>
    <p id="message">Texte initial</p>
    <button id="changeBtn">Changer le texte</button>
  </div>
  <script src="script.js"></script>
```

```
</body>
</html>
```

📄 **style.css**

```
.container {
  max-width: 600px;
  margin: 50px auto;
  padding: 30px;
  background: white;
  border-radius: 10px;
  box-shadow: 0 4px 6px rgba(0,0,0,0.1);
  text-align: center;
}

h1 {
  color: #1F2937;
  margin-bottom: 20px;
}

#message {
  font-size: 20px;
  color: #6B7280;
  margin: 30px 0;
}

#changeBtn {
  background: #3B82F6;
```

```
  color: white;
  border: none;
  padding: 12px 24px;
  border-radius: 6px;
  cursor: pointer;
  font-size: 16px;
  transition: background 0.3s;
}

#changeBtn:hover {
  background: #2563EB;
}
```

📄 **script.js**

```
const message = document.getElementById('message');
const changeBtn = document.getElementById('changeBtn');

changeBtn.addEventListener('click', () => {
  message.textContent = 'Le texte a été changé avec JavaScript !';
  message.style.color = '#10B981';
  message.style.fontWeight = 'bold';
});
```

PARTIE 2 — Composants Interactifs

2.1 Menu déroulant (Dropdown)

Créons un menu qui s'ouvre et se ferme au clic.

📄 **dropdown.html**

```
<div class="dropdown">
  <button class="dropdown-btn" id="dropdownBtn">
    Menu ▼
  </button>
  <div class="dropdown-content" id="dropdownContent">
    <a href="#">Accueil</a>
    <a href="#">Services</a>
    <a href="#">À propos</a>
    <a href="#">Contact</a>
  </div>
</div>
```

📄 **dropdown.css**

```
.dropdown {
  position: relative;
  display: inline-block;
}

.dropdown-btn {
  background: #3B82F6;
  color: white;
```

```
  padding: 12px 20px;
  border: none;
  border-radius: 6px;
  cursor: pointer;
  font-size: 16px;
}

.dropdown-content {
  display: none; /* Caché par défaut */
  position: absolute;
  background: white;
  min-width: 200px;
  box-shadow: 0 8px 16px rgba(0,0,0,0.2);
  border-radius: 6px;
  margin-top: 5px;
  z-index: 1;
}

.dropdown-content.show {
  display: block; /* Visible quand classe 'show' */
}

.dropdown-content a {
  color: #1F2937;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}
```

```
.dropdown-content a:hover {
  background: #F3F4F6;
}
```

📄 **dropdown.js**

```
const dropdownBtn = document.getElementById('dropdownBtn');
const dropdownContent = document.getElementById('dropdownContent');

// Toggle (ouvrir/fermer) au clic
dropdownBtn.addEventListener('click', () => {
  dropdownContent.classList.toggle('show');
});

// Fermer si clic ailleurs sur la page
window.addEventListener('click', (e) => {
  if (!e.target.matches('.dropdown-btn')) {
    if (dropdownContent.classList.contains('show')) {
      dropdownContent.classList.remove('show');
    }
  }
});
```

💡 **Astuce**

La méthode classList.toggle() ajoute la classe si elle n'existe pas, et la retire si elle existe. Parfait pour les éléments on/off !

2.2 Modal / Popup

📄 **modal.html**

```
<button id="openModal">Ouvrir la popup</button>

<div class="modal" id="myModal">
  <div class="modal-content">
    <span class="close" id="closeModal">&times;</span>
    <h2>Titre de la popup</h2>
    <p>Contenu de la popup ici...</p>
  </div>
</div>
```

📄 **modal.css**

```
.modal {
  display: none; /* Caché par défaut */
  position: fixed;
  z-index: 1000;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.5); /* Fond sombre */
}

.modal.show {
  display: flex;
```

```
  align-items: center;
  justify-content: center;
}

.modal-content {
  background: white;
  padding: 30px;
  border-radius: 10px;
  max-width: 500px;
  width: 90%;
  position: relative;
}

.close {
  position: absolute;
  right: 15px;
  top: 10px;
  font-size: 28px;
  font-weight: bold;
  color: #999;
  cursor: pointer;
}

.close:hover {
  color: #333;
}
```

📄 **modal.js**

```
const openModal = document.getElementById('openModal');
const closeModal = document.getElementById('closeModal');
const modal = document.getElementById('myModal');

// Ouvrir la modal
openModal.addEventListener('click', () => {
  modal.classList.add('show');
});

// Fermer la modal
closeModal.addEventListener('click', () => {
  modal.classList.remove('show');
});

// Fermer si clic sur le fond sombre
modal.addEventListener('click', (e) => {
  if (e.target === modal) {
    modal.classList.remove('show');
  }
});
```

✏️ EXERCICE 2 : Galerie avec modal

Créez une galerie de 4 images. Au clic sur une image, ouvrez une modal qui affiche l'image en grand avec sa description.

PARTIE 3 : FORMULAIRES ET VALIDATION

Les formulaires sont essentiels pour tout site interactif. Vous allez apprendre à créer des formulaires complets avec validation JavaScript

EXERCICE 7 — Formulaire de Contact Simple

Créons un formulaire de contact avec validation en temps réel.

HTML - Structure du formulaire

📄 **index.html - Exercice 7**

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Formulaire de Contact</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <div class="container">
    <h1>📧 Contactez-nous</h1>
    <p class="subtitle">Remplissez le formulaire ci-dessous</p>

    <form id="contactForm">

      <!-- Champ Nom -->
      <div class="form-group">
        <label for="name">Nom complet *</label>
        <input type="text" id="name" name="name" required>
        <span class="error" id="nameError"></span>
      </div>

      <!-- Champ Email -->
      <div class="form-group">
        <label for="email">Email *</label>
        <input type="email" id="email" name="email" required>
        <span class="error" id="emailError"></span>
      </div>

      <!-- Champ Sujet -->
      <div class="form-group">
        <label for="subject">Sujet *</label>
        <select id="subject" name="subject" required>
          <option value="">Choisir un sujet</option>
          <option value="question">Question générale</option>
          <option value="support">Support technique</option>
```

```
          <option value="feedback">Feedback</option>
          <option value="other">Autre</option>
        </select>
        <span class="error" id="subjectError"></span>
      </div>

      <!-- Champ Message -->
      <div class="form-group">
        <label for="message">Message *</label>
        <textarea id="message" name="message" rows="5" required></textarea>
        <span class="error" id="messageError"></span>
      </div>

      <!-- Case à cocher -->
      <div class="form-group checkbox-group">
        <input type="checkbox" id="terms" name="terms" required>
        <label for="terms">
          J'accepte les conditions d'utilisation *
        </label>
        <span class="error" id="termsError"></span>
      </div>

      <!-- Bouton d'envoi -->
      <button type="submit" class="submit-btn">Envoyer le message</button>

      <!-- Message de succès -->
      <div class="success-message" id="successMessage">
        ✅ Message envoyé avec succès !
      </div>

    </form>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

CSS - Style du formulaire

📄 **style.css - Exercice 7**

```css
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  background: linear-gradient(135deg, #10B981 0%, #059669 100%);
  min-height: 100vh;
  padding: 40px 20px;
}

.container {
  max-width: 600px;
  margin: 0 auto;
  background: white;
  padding: 40px;
  border-radius: 20px;
  box-shadow: 0 10px 40px rgba(0, 0, 0, 0.2);
}

h1 {
  color: #10B981;
  text-align: center;
  margin-bottom: 10px;
  font-size: 32px;
}

.subtitle {
  text-align: center;
  color: #6B7280;
  margin-bottom: 30px;
}

/* Groupes de formulaire */
.form-group {
  margin-bottom: 25px;
}
```

```
label {
  display: block;
  margin-bottom: 8px;
  color: #374151;
  font-weight: 600;
  font-size: 14px;
}

/* Champs de saisie */
input[type="text"],
input[type="email"],
select,
textarea {
  width: 100%;
  padding: 12px 15px;
  border: 2px solid #E5E7EB;
  border-radius: 8px;
  font-size: 16px;
  font-family: inherit;
  transition: all 0.3s;
}

input:focus,
select:focus,
textarea:focus {
  outline: none;
  border-color: #10B981;
  box-shadow: 0 0 0 3px rgba(16, 185, 129, 0.1);
}

/* Textarea */
textarea {
  resize: vertical;
  min-height: 120px;
}

/* Case à cocher */
.checkbox-group {
  display: flex;
  align-items: center;
  gap: 10px;
}
```

```css
.checkbox-group input[type="checkbox"] {
  width: auto;
  cursor: pointer;
}

.checkbox-group label {
  margin: 0;
  cursor: pointer;
  font-weight: normal;
}

/* Messages d'erreur */
.error {
  display: block;
  color: #EF4444;
  font-size: 13px;
  margin-top: 5px;
  min-height: 18px;
}

/* Champs invalides */
input.invalid,
select.invalid,
textarea.invalid {
  border-color: #EF4444;
  background-color: #FEF2F2;
}

/* Champs valides */
input.valid,
select.valid,
textarea.valid {
  border-color: #10B981;
  background-color: #F0FDF4;
}

/* Bouton d'envoi */
.submit-btn {
  width: 100%;
  padding: 15px;
  background: #10B981;
  color: white;
  border: none;
```

```
  border-radius: 10px;
  font-size: 18px;
  font-weight: 600;
  cursor: pointer;
  transition: all 0.3s;
  margin-top: 10px;
}

.submit-btn:hover {
  background: #059669;
  transform: translateY(-2px);
  box-shadow: 0 4px 12px rgba(16, 185, 129, 0.3);
}

.submit-btn:active {
  transform: translateY(0);
}

/* Message de succès */
.success-message {
  display: none;
  margin-top: 20px;
  padding: 15px;
  background: #D1FAE5;
  color: #065F46;
  border-radius: 8px;
  text-align: center;
  font-weight: 600;
}

.success-message.show {
  display: block;
  animation: slideDown 0.3s ease;
}

@keyframes slideDown {
  from {
    opacity: 0;
    transform: translateY(-10px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
```

```
  }
}
```

JavaScript - Validation du formulaire

📄 **script.js - Exercice 7**

```
// Récupérer les éléments du formulaire
const form = document.getElementById('contactForm');
const nameInput = document.getElementById('name');
const emailInput = document.getElementById('email');
const subjectSelect = document.getElementById('subject');
const messageTextarea = document.getElementById('message');
const termsCheckbox = document.getElementById('terms');
const successMessage = document.getElementById('successMessage');

// Fonction de validation du nom
function validateName() {
  const nameValue = nameInput.value.trim();
  const nameError = document.getElementById('nameError');

  if (nameValue === '') {
    nameError.textContent = 'Le nom est requis';
    nameInput.classList.add('invalid');
    nameInput.classList.remove('valid');
    return false;
  } else if (nameValue.length < 2) {
    nameError.textContent = 'Le nom doit contenir au moins 2 caractères';
    nameInput.classList.add('invalid');
    nameInput.classList.remove('valid');
    return false;
  } else {
    nameError.textContent = '';
    nameInput.classList.remove('invalid');
    nameInput.classList.add('valid');
    return true;
  }
}

// Fonction de validation de l'email
function validateEmail() {
  const emailValue = emailInput.value.trim();
  const emailError = document.getElementById('emailError');
  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

  if (emailValue === '') {
    emailError.textContent = 'L\'email est requis';
    emailInput.classList.add('invalid');
```

```
    emailInput.classList.remove('valid');
    return false;
  } else if (!emailRegex.test(emailValue)) {
    emailError.textContent = 'Email invalide (ex: nom@example.com)';
    emailInput.classList.add('invalid');
    emailInput.classList.remove('valid');
    return false;
  } else {
    emailError.textContent = '';
    emailInput.classList.remove('invalid');
    emailInput.classList.add('valid');
    return true;
  }
}

// Fonction de validation du sujet
function validateSubject() {
  const subjectValue = subjectSelect.value;
  const subjectError = document.getElementById('subjectError');

  if (subjectValue === '') {
    subjectError.textContent = 'Veuillez choisir un sujet';
    subjectSelect.classList.add('invalid');
    subjectSelect.classList.remove('valid');
    return false;
  } else {
    subjectError.textContent = '';
    subjectSelect.classList.remove('invalid');
    subjectSelect.classList.add('valid');
    return true;
  }
}

// Fonction de validation du message
function validateMessage() {
  const messageValue = messageTextarea.value.trim();
  const messageError = document.getElementById('messageError');

  if (messageValue === '') {
    messageError.textContent = 'Le message est requis';
    messageTextarea.classList.add('invalid');
    messageTextarea.classList.remove('valid');
    return false;
```

```
  } else if (messageValue.length < 10) {
    messageError.textContent = 'Le message doit contenir au moins 10 caractères';
    messageTextarea.classList.add('invalid');
    messageTextarea.classList.remove('valid');
    return false;
  } else {
    messageError.textContent = '';
    messageTextarea.classList.remove('invalid');
    messageTextarea.classList.add('valid');
    return true;
  }
}

// Fonction de validation des conditions
function validateTerms() {
  const termsError = document.getElementById('termsError');

  if (!termsCheckbox.checked) {
    termsError.textContent = 'Vous devez accepter les conditions';
    return false;
  } else {
    termsError.textContent = '';
    return true;
  }
}

// Validation en temps réel
nameInput.addEventListener('blur', validateName);
emailInput.addEventListener('blur', validateEmail);
subjectSelect.addEventListener('change', validateSubject);
messageTextarea.addEventListener('blur', validateMessage);
termsCheckbox.addEventListener('change', validateTerms);

// Soumission du formulaire
form.addEventListener('submit', function(e) {
  e.preventDefault();

  // Valider tous les champs
  const isNameValid = validateName();
  const isEmailValid = validateEmail();
  const isSubjectValid = validateSubject();
  const isMessageValid = validateMessage();
  const areTermsValid = validateTerms();
```

```
  // Si tout est valide
  if (isNameValid && isEmailValid && isSubjectValid && isMessageValid &&
areTermsValid) {
    // Afficher le message de succès
    successMessage.classList.add('show');

    // Réinitialiser le formulaire après 2 secondes
    setTimeout(() => {
      form.reset();
      successMessage.classList.remove('show');
      // Retirer les classes de validation
      document.querySelectorAll('.valid').forEach(el => {
        el.classList.remove('valid');
      });
    }, 2000);

    // Ici, vous pouvez envoyer les données à un serveur
    console.log('Formulaire soumis avec succès !');
    console.log({
      name: nameInput.value,
      email: emailInput.value,
      subject: subjectSelect.value,
      message: messageTextarea.value
    });
  }
});
```

💡 FONCTIONNALITÉS

✅ Validation en temps réel (quand on quitte un champ)

✅ Messages d'erreur spécifiques pour chaque champ

✅ Vérification format email avec regex

✅ Longueur minimale pour nom et message

✅ Cases à cocher obligatoires

✅ Animation du message de succès

EXERCICE 8 — Formulaire d'Inscription avec Confirmation Mot de Passe

Créons un formulaire d'inscription complet avec vérification de la force du mot de passe.

HTML - Formulaire d'inscription

📄 **index.html - Exercice 8**

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Inscription</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <div class="container">
    <h1>🔐 Créer un compte</h1>

    <form id="signupForm">

      <div class="form-group">
        <label for="username">Nom d'utilisateur *</label>
        <input type="text" id="username" required>
        <span class="error" id="usernameError"></span>
      </div>

      <div class="form-group">
        <label for="email">Email *</label>
        <input type="email" id="email" required>
        <span class="error" id="emailError"></span>
      </div>

      <div class="form-group">
        <label for="password">Mot de passe *</label>
        <input type="password" id="password" required>
        <div class="password-strength">
          <div class="strength-bar" id="strengthBar"></div>
        </div>
        <span class="strength-text" id="strengthText"></span>
        <span class="error" id="passwordError"></span>
      </div>
```

```
      <div class="form-group">
        <label for="confirmPassword">Confirmer le mot de passe *</label>
        <input type="password" id="confirmPassword" required>
        <span class="error" id="confirmError"></span>
      </div>

      <div class="form-group checkbox-group">
        <input type="checkbox" id="newsletter">
        <label for="newsletter">Recevoir la newsletter</label>
      </div>

      <button type="submit" class="submit-btn">S'inscrire</button>

      <div class="success-message" id="successMessage">
        ✅ Compte créé avec succès !
      </div>

    </form>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

CSS - Style avec indicateur de force

📄 **style.css - Ajout Exercice 8**

```css
/* Reprenez le CSS de l'exercice 7 et ajoutez : */

/* Barre de force du mot de passe */
.password-strength {
  width: 100%;
  height: 8px;
  background: #E5E7EB;
  border-radius: 4px;
  margin-top: 8px;
  overflow: hidden;
}

.strength-bar {
  height: 100%;
  width: 0;
  transition: all 0.3s ease;
  border-radius: 4px;
}

/* Couleurs selon la force */
.strength-bar.weak {
  width: 33%;
  background: #EF4444;
}

.strength-bar.medium {
  width: 66%;
  background: #F59E0B;
}

.strength-bar.strong {
  width: 100%;
  background: #10B981;
}

/* Texte de force */
.strength-text {
  display: block;
  font-size: 12px;
  margin-top: 5px;
```

```
  font-weight: 600;
}

.strength-text.weak {
  color: #EF4444;
}

.strength-text.medium {
  color: #F59E0B;
}

.strength-text.strong {
  color: #10B981;
}
```

JavaScript - Validation avec force du mot de passe

📄 **script.js - Exercice 8**

```
const form = document.getElementById('signupForm');
const usernameInput = document.getElementById('username');
const emailInput = document.getElementById('email');
const passwordInput = document.getElementById('password');
const confirmPasswordInput = document.getElementById('confirmPassword');
const strengthBar = document.getElementById('strengthBar');
const strengthText = document.getElementById('strengthText');

// Validation du nom d'utilisateur
function validateUsername() {
  const value = usernameInput.value.trim();
  const error = document.getElementById('usernameError');

  if (value === '') {
    error.textContent = 'Nom d\'utilisateur requis';
    return false;
  } else if (value.length < 3) {
    error.textContent = 'Minimum 3 caractères';
    return false;
  } else if (!/^[a-zA-Z0-9_]+$/.test(value)) {
    error.textContent = 'Lettres, chiffres et underscore uniquement';
    return false;
  } else {
    error.textContent = '';
    return true;
  }
}

// Vérifier la force du mot de passe
function checkPasswordStrength() {
  const password = passwordInput.value;
  let strength = 0;

  // Longueur
  if (password.length >= 8) strength++;
  if (password.length >= 12) strength++;

  // Contient des minuscules
  if (/[a-z]/.test(password)) strength++;

  // Contient des majuscules
```

```
  if (/[A-Z]/.test(password)) strength++;

  // Contient des chiffres
  if (/[0-9]/.test(password)) strength++;

  // Contient des caractères spéciaux
  if (/[^a-zA-Z0-9]/.test(password)) strength++;

  // Mettre à jour l'affichage
  strengthBar.className = 'strength-bar';
  strengthText.className = 'strength-text';

  if (strength <= 2) {
    strengthBar.classList.add('weak');
    strengthText.classList.add('weak');
    strengthText.textContent = '❌ Faible';
  } else if (strength <= 4) {
    strengthBar.classList.add('medium');
    strengthText.classList.add('medium');
    strengthText.textContent = '⚠ Moyen';
  } else {
    strengthBar.classList.add('strong');
    strengthText.classList.add('strong');
    strengthText.textContent = '✅ Fort';
  }
}

// Validation du mot de passe
function validatePassword() {
  const value = passwordInput.value;
  const error = document.getElementById('passwordError');

  if (value === '') {
    error.textContent = 'Mot de passe requis';
    return false;
  } else if (value.length < 8) {
    error.textContent = 'Minimum 8 caractères';
    return false;
  } else {
    error.textContent = '';
    return true;
  }
}
```

```
// Validation de la confirmation
function validateConfirmPassword() {
  const password = passwordInput.value;
  const confirm = confirmPasswordInput.value;
  const error = document.getElementById('confirmError');

  if (confirm === '') {
    error.textContent = 'Confirmation requise';
    return false;
  } else if (password !== confirm) {
    error.textContent = 'Les mots de passe ne correspondent pas';
    return false;
  } else {
    error.textContent = '';
    return true;
  }
}

// Événements
passwordInput.addEventListener('input', checkPasswordStrength);
passwordInput.addEventListener('blur', validatePassword);
confirmPasswordInput.addEventListener('blur', validateConfirmPassword);
usernameInput.addEventListener('blur', validateUsername);

form.addEventListener('submit', function(e) {
  e.preventDefault();

  if (validateUsername() && validatePassword() && validateConfirmPassword()) {
    document.getElementById('successMessage').classList.add('show');
    setTimeout(() => form.reset(), 2000);
  }
});
```

💡 CRITÈRES DE FORCE DU MOT DE PASSE

Le script vérifie 6 critères :

1. Longueur >= 8 caractères
2. Longueur >= 12 caractères
3. Contient des minuscules (a-z)
4. Contient des majuscules (A-Z)
5. Contient des chiffres (0-9)
6. Contient des caractères spéciaux (!@#$%...)

PARTIE 4 : PROJETS D'INTÉGRATION AVANCÉS

Dans cette partie, vous allez créer des projets complets qui combinent HTML, CSS et JavaScript de manière professionnelle.

EXERCICE 9 — Accordéon FAQ Interactif

Créons une section FAQ avec des accordéons qui s'ouvrent et se ferment.

HTML - Structure FAQ

📄 **index.html - Exercice 9**

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>FAQ Accordéon</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <div class="container">
    <h1>❓ Questions Fréquentes</h1>
    <p class="subtitle">Cliquez sur une question pour voir la réponse</p>

    <div class="accordion">

      <!-- Question 1 -->
      <div class="accordion-item">
        <button class="accordion-header">
          <span>Qu'est-ce que le développement web ?</span>
          <span class="icon">+</span>
        </button>
        <div class="accordion-content">
          <p>
            Le développement web consiste à créer des sites internet
            et des applications web en utilisant HTML, CSS et JavaScript.
            C'est un domaine passionnant qui évolue constamment.
          </p>
        </div>
      </div>

      <!-- Question 2 -->
      <div class="accordion-item">
        <button class="accordion-header">
          <span>Combien de temps pour apprendre ?</span>
          <span class="icon">+</span>
        </button>
        <div class="accordion-content">
          <p>
```

```
        Avec de la pratique régulière, vous pouvez maîtriser les bases
        en 2-3 mois. Devenir expert prend plusieurs années, mais vous
        pouvez créer des projets sympas dès les premiers mois !
      </p>
    </div>
  </div>

  <!-- Question 3 -->
  <div class="accordion-item">
    <button class="accordion-header">
      <span>Quels sont les outils nécessaires ?</span>
      <span class="icon">+</span>
    </button>
    <div class="accordion-content">
      <p>
        Vous avez besoin de :
      </p>
      <ul>
        <li>Un éditeur de code (VS Code recommandé)</li>
        <li>Un navigateur web moderne (Chrome, Firefox)</li>
        <li>De la motivation et de la curiosité !</li>
      </ul>
    </div>
  </div>

  <!-- Question 4 -->
  <div class="accordion-item">
    <button class="accordion-header">
      <span>Faut-il connaître les mathématiques ?</span>
      <span class="icon">+</span>
    </button>
    <div class="accordion-content">
      <p>
        Pas besoin d'être un génie en maths ! Les bases de logique
        suffisent. Le développement web est plus créatif que mathématique.
      </p>
    </div>
  </div>

  <!-- Question 5 -->
  <div class="accordion-item">
    <button class="accordion-header">
      <span>Peut-on gagner sa vie avec le web ?</span>
      <span class="icon">+</span>
```

```
        </button>
        <div class="accordion-content">
          <p>
            Absolument ! Le développement web offre de nombreuses
            opportunités : emplois en entreprise, freelance, création
            de startups, etc. C'est un secteur très demandé.
          </p>
        </div>
      </div>

    </div>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

CSS - Style de l'accordéon

📄 **style.css - Exercice 9**

```css
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: Arial, sans-serif;
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  min-height: 100vh;
  padding: 40px 20px;
}

.container {
  max-width: 800px;
  margin: 0 auto;
}

h1 {
  color: white;
  text-align: center;
  margin-bottom: 10px;
  font-size: 42px;
}

.subtitle {
  text-align: center;
  color: rgba(255, 255, 255, 0.9);
  margin-bottom: 40px;
  font-size: 18px;
}

/* Accordéon */
.accordion {
  background: white;
  border-radius: 15px;
  overflow: hidden;
  box-shadow: 0 10px 40px rgba(0, 0, 0, 0.2);
}
```

```
.accordion-item {
  border-bottom: 1px solid #E5E7EB;
}

.accordion-item:last-child {
  border-bottom: none;
}

/* En-tête (question) */
.accordion-header {
  width: 100%;
  padding: 20px 25px;
  background: white;
  border: none;
  display: flex;
  justify-content: space-between;
  align-items: center;
  cursor: pointer;
  font-size: 18px;
  font-weight: 600;
  color: #1F2937;
  text-align: left;
  transition: all 0.3s;
}

.accordion-header:hover {
  background: #F9FAFB;
}

.accordion-header.active {
  background: #EDE9FE;
  color: #667eea;
}

/* Icône + */
.icon {
  font-size: 28px;
  font-weight: bold;
  color: #667eea;
  transition: transform 0.3s;
}

.accordion-header.active .icon {
```

```
  transform: rotate(45deg);
}

/* Contenu (réponse) */
.accordion-content {
  max-height: 0;
  overflow: hidden;
  transition: max-height 0.3s ease;
  background: #F9FAFB;
}

.accordion-content p,
.accordion-content ul {
  padding: 20px 25px;
  color: #4B5563;
  line-height: 1.6;
}

.accordion-content ul {
  list-style-position: inside;
}

.accordion-content li {
  margin-bottom: 8px;
}
```

JavaScript - Animation de l'accordéon

📄 **script.js - Exercice 9**

```
// Récupérer tous les en-têtes d'accordéon
const accordionHeaders = document.querySelectorAll('.accordion-header');

accordionHeaders.forEach(header => {
  header.addEventListener('click', function() {

    // Récupérer le contenu associé
    const content = this.nextElementSibling;

    // Vérifier si cet item est déjà actif
    const isActive = this.classList.contains('active');

    // Fermer tous les accordéons
    accordionHeaders.forEach(h => {
      h.classList.remove('active');
      h.nextElementSibling.style.maxHeight = null;
    });

    // Si l'item n'était pas actif, l'ouvrir
    if (!isActive) {
      this.classList.add('active');
      content.style.maxHeight = content.scrollHeight + 'px';
    }

  });
});
```

💡 COMMENT ÇA MARCHE ?

1. On écoute le clic sur chaque en-tête
2. On ferme tous les accordéons ouverts
3. Si l'item cliqué n'était pas ouvert, on l'ouvre
4. L'animation se fait avec max-height et scrollHeight
5. L'icône + tourne de 45° pour devenir ×

EXERCICE 10 — Fenêtre Modale (Popup)

Créons une fenêtre modale moderne qui s'ouvre au-dessus du contenu.

📄 **Code complet - Exercice 10**

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Modale</title>
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    body {
      font-family: Arial, sans-serif;
      padding: 50px;
      background: #f5f5f5;
    }

    h1 {
      color: #667eea;
      margin-bottom: 20px;
    }

    /* Bouton d'ouverture */
    .open-modal-btn {
      padding: 15px 30px;
      background: #667eea;
      color: white;
      border: none;
      border-radius: 8px;
      font-size: 16px;
      cursor: pointer;
      transition: all 0.3s;
    }

    .open-modal-btn:hover {
      background: #5568d3;
      transform: translateY(-2px);
    }
```

```css
/* Overlay (fond sombre) */
.modal-overlay {
  display: none;
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.7);
  z-index: 1000;
  animation: fadeIn 0.3s;
}

.modal-overlay.active {
  display: flex;
  justify-content: center;
  align-items: center;
}

/* Modale */
.modal {
  background: white;
  border-radius: 15px;
  width: 90%;
  max-width: 500px;
  padding: 40px;
  position: relative;
  animation: slideDown 0.3s;
}

/* Bouton fermeture */
.close-btn {
  position: absolute;
  top: 15px;
  right: 15px;
  background: #EF4444;
  color: white;
  border: none;
  width: 35px;
  height: 35px;
  border-radius: 50%;
  cursor: pointer;
  font-size: 20px;
```

```
    }

    .modal h2 {
      color: #667eea;
      margin-bottom: 15px;
    }

    .modal p {
      color: #6B7280;
      line-height: 1.6;
      margin-bottom: 20px;
    }

    /* Animations */
    @keyframes fadeIn {
      from { opacity: 0; }
      to { opacity: 1; }
    }

    @keyframes slideDown {
      from {
        opacity: 0;
        transform: translateY(-50px);
      }
      to {
        opacity: 1;
        transform: translateY(0);
      }
    }
  </style>
</head>
<body>

  <h1>Fenêtre Modale</h1>
  <p>Cliquez sur le bouton pour ouvrir la modale</p>
  <button class="open-modal-btn" id="openModal">Ouvrir la modale</button>

  <!-- Modale -->
  <div class="modal-overlay" id="modalOverlay">
    <div class="modal">
      <button class="close-btn" id="closeModal">&times;</button>
      <h2>🎉 Bienvenue !</h2>
      <p>
```

```
        Ceci est une fenêtre modale. Elle apparaît au-dessus du contenu
        principal et attire l'attention de l'utilisateur.
      </p>
      <p>
        Vous pouvez la fermer en cliquant sur le bouton X, en appuyant
        sur Échap, ou en cliquant en dehors de la modale.
      </p>
    </div>
  </div>

  <script>
    const openBtn = document.getElementById('openModal');
    const closeBtn = document.getElementById('closeModal');
    const overlay = document.getElementById('modalOverlay');

    // Ouvrir la modale
    openBtn.addEventListener('click', () => {
      overlay.classList.add('active');
    });

    // Fermer la modale (bouton X)
    closeBtn.addEventListener('click', () => {
      overlay.classList.remove('active');
    });

    // Fermer en cliquant sur l'overlay
    overlay.addEventListener('click', (e) => {
      if (e.target === overlay) {
        overlay.classList.remove('active');
      }
    });

    // Fermer avec la touche Échap
    document.addEventListener('keydown', (e) => {
      if (e.key === 'Escape') {
        overlay.classList.remove('active');
      }
    });
  </script>

</body>
</html>
```

💡 FONCTIONNALITÉS MODALE

✅ S'ouvre au centre de l'écran
✅ Fond sombre (overlay)
✅ Animations fluides
✅ Fermeture multiple : bouton X, clic dehors, touche Échap
✅ Position fixe (suit le scroll)

EXERCICE 11 — Carrousel d'Images

Créons un carrousel d'images avec navigation par boutons et indicateurs.

📄 **Code complet - Exercice 11**

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Carrousel</title>
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    body {
      font-family: Arial, sans-serif;
      background: #1F2937;
      padding: 50px 20px;
    }

    h1 {
      color: white;
      text-align: center;
      margin-bottom: 40px;
    }

    /* Conteneur du carrousel */
    .carousel-container {
      max-width: 800px;
      margin: 0 auto;
      position: relative;
      background: white;
      border-radius: 15px;
      overflow: hidden;
      box-shadow: 0 10px 40px rgba(0,0,0,0.3);
    }

    /* Zone des slides */
    .carousel-slides {
      display: flex;
      transition: transform 0.5s ease;
    }
```

```
.slide {
  min-width: 100%;
  height: 500px;
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 48px;
  font-weight: bold;
  color: white;
}

.slide:nth-child(1) { background: #EF4444; }
.slide:nth-child(2) { background: #10B981; }
.slide:nth-child(3) { background: #3B82F6; }
.slide:nth-child(4) { background: #F59E0B; }
.slide:nth-child(5) { background: #8B5CF6; }

/* Boutons précédent/suivant */
.carousel-btn {
  position: absolute;
  top: 50%;
  transform: translateY(-50%);
  background: rgba(0,0,0,0.5);
  color: white;
  border: none;
  width: 50px;
  height: 50px;
  border-radius: 50%;
  cursor: pointer;
  font-size: 24px;
  transition: all 0.3s;
  z-index: 10;
}

.carousel-btn:hover {
  background: rgba(0,0,0,0.8);
}

.prev-btn { left: 20px; }
.next-btn { right: 20px; }

/* Indicateurs (points) */
```

```
    .carousel-indicators {
      position: absolute;
      bottom: 20px;
      left: 50%;
      transform: translateX(-50%);
      display: flex;
      gap: 10px;
    }

    .indicator {
      width: 12px;
      height: 12px;
      border-radius: 50%;
      background: rgba(255,255,255,0.5);
      cursor: pointer;
      transition: all 0.3s;
    }

    .indicator.active {
      background: white;
      width: 30px;
      border-radius: 6px;
    }
  </style>
</head>
<body>

  <h1>🎠 Carrousel d'Images</h1>

  <div class="carousel-container">
    <div class="carousel-slides" id="carouselSlides">
      <div class="slide">Slide 1</div>
      <div class="slide">Slide 2</div>
      <div class="slide">Slide 3</div>
      <div class="slide">Slide 4</div>
      <div class="slide">Slide 5</div>
    </div>

    <button class="carousel-btn prev-btn" id="prevBtn">&lt;</button>
    <button class="carousel-btn next-btn" id="nextBtn">&gt;</button>

    <div class="carousel-indicators" id="indicators"></div>
  </div>
```

```
<script>
  const slides = document.getElementById('carouselSlides');
  const prevBtn = document.getElementById('prevBtn');
  const nextBtn = document.getElementById('nextBtn');
  const indicatorsContainer = document.getElementById('indicators');

  let currentIndex = 0;
  const totalSlides = document.querySelectorAll('.slide').length;

  // Créer les indicateurs
  for (let i = 0; i < totalSlides; i++) {
    const indicator = document.createElement('div');
    indicator.classList.add('indicator');
    if (i === 0) indicator.classList.add('active');
    indicator.addEventListener('click', () => goToSlide(i));
    indicatorsContainer.appendChild(indicator);
  }

  function updateCarousel() {
    slides.style.transform = `translateX(-${currentIndex * 100}%)`;

    // Mettre à jour les indicateurs
    document.querySelectorAll('.indicator').forEach((ind, i) => {
      ind.classList.toggle('active', i === currentIndex);
    });
  }

  function goToSlide(index) {
    currentIndex = index;
    updateCarousel();
  }

  function nextSlide() {
    currentIndex = (currentIndex + 1) % totalSlides;
    updateCarousel();
  }

  function prevSlide() {
    currentIndex = (currentIndex - 1 + totalSlides) % totalSlides;
    updateCarousel();
  }
```

```
    nextBtn.addEventListener('click', nextSlide);
    prevBtn.addEventListener('click', prevSlide);

    // Auto-play (optionnel)
    // setInterval(nextSlide, 3000);
  </script>

</body>
</html>
```

EXERCICE 12 — Galerie avec Filtres Dynamiques

Créons une galerie d'images filtrables par catégorie.

📄 **Code complet - Exercice 12**

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Galerie Filtrée</title>
  <style>
    * { margin: 0; padding: 0; box-sizing: border-box; }

    body {
      font-family: Arial, sans-serif;
      background: #f5f5f5;
      padding: 40px 20px;
    }

    h1 {
      text-align: center;
      color: #667eea;
      margin-bottom: 30px;
    }

    /* Boutons de filtre */
    .filter-buttons {
      display: flex;
      justify-content: center;
      gap: 15px;
      margin-bottom: 40px;
      flex-wrap: wrap;
    }

    .filter-btn {
      padding: 12px 25px;
      background: white;
      border: 2px solid #E5E7EB;
      border-radius: 25px;
      cursor: pointer;
      font-size: 16px;
      transition: all 0.3s;
    }
```

```
.filter-btn:hover {
  border-color: #667eea;
  color: #667eea;
}

.filter-btn.active {
  background: #667eea;
  color: white;
  border-color: #667eea;
}

/* Grille de la galerie */
.gallery {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
  gap: 20px;
  max-width: 1200px;
  margin: 0 auto;
}

.gallery-item {
  background: white;
  border-radius: 10px;
  overflow: hidden;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
  transition: all 0.3s;
}

.gallery-item:hover {
  transform: translateY(-5px);
  box-shadow: 0 5px 20px rgba(0,0,0,0.2);
}

.gallery-item.hide {
  display: none;
}

.item-image {
  width: 100%;
  height: 200px;
  display: flex;
  justify-content: center;
  align-items: center;
```

```
      font-size: 48px;
    }

    .item-info {
      padding: 15px;
    }

    .item-title {
      font-weight: bold;
      color: #1F2937;
      margin-bottom: 5px;
    }

    .item-category {
      color: #6B7280;
      font-size: 14px;
    }
  </style>
</head>
<body>

  <h1>🖼️ Galerie Filtrée</h1>

  <!-- Boutons de filtre -->
  <div class="filter-buttons">
    <button class="filter-btn active" data-filter="all">Tout</button>
    <button class="filter-btn" data-filter="nature">Nature</button>
    <button class="filter-btn" data-filter="ville">Ville</button>
    <button class="filter-btn" data-filter="animaux">Animaux</button>
  </div>

  <!-- Galerie -->
  <div class="gallery" id="gallery">
    <!-- Les items seront générés en JavaScript -->
  </div>

  <script>
    // Données de la galerie
    const galleryData = [
      { title: 'Montagne', category: 'nature', emoji: '🏔️', color: '#10B981' },
      { title: 'Chat', category: 'animaux', emoji: '🐱', color: '#F59E0B' },
      { title: 'Gratte-ciel', category: 'ville', emoji: '🏙️', color: '#3B82F6' },
      { title: 'Forêt', category: 'nature', emoji: '🌲', color: '#059669' },
```

```
  { title: 'Chien', category: 'animaux', emoji: '🐕', color: '#EF4444' },
  { title: 'Rue', category: 'ville', emoji: '🛣️', color: '#6366F1' },
  { title: 'Plage', category: 'nature', emoji: '🏖️', color: '#14B8A6' },
  { title: 'Oiseau', category: 'animaux', emoji: '🦜', color: '#8B5CF6' },
  { title: 'Pont', category: 'ville', emoji: '🌉', color: '#EC4899' },
];

const gallery = document.getElementById('gallery');
const filterButtons = document.querySelectorAll('.filter-btn');

// Générer les items de la galerie
galleryData.forEach(item => {
  const itemDiv = document.createElement('div');
  itemDiv.className = 'gallery-item';
  itemDiv.dataset.category = item.category;

  itemDiv.innerHTML = `
    <div class="item-image" style="background: ${item.color}">
      ${item.emoji}
    </div>
    <div class="item-info">
      <div class="item-title">${item.title}</div>
      <div class="item-category">${item.category}</div>
    </div>
  `;

  gallery.appendChild(itemDiv);
});

// Filtrage
filterButtons.forEach(button => {
  button.addEventListener('click', function() {
    const filter = this.dataset.filter;

    // Mettre à jour les boutons actifs
    filterButtons.forEach(btn => btn.classList.remove('active'));
    this.classList.add('active');

    // Filtrer les items
    const items = document.querySelectorAll('.gallery-item');
    items.forEach(item => {
      if (filter === 'all' || item.dataset.category === filter) {
        item.classList.remove('hide');
```

```
          } else {
            item.classList.add('hide');
          }
        });
      });
    });
  </script>

</body>
</html>
```

💡 CONCEPTS UTILISÉS

✅ Grid CSS pour la mise en page responsive

✅ data-attributes pour stocker la catégorie

✅ Génération dynamique du HTML avec JavaScript

✅ Filtrage en ajoutant/retirant la classe 'hide'

✅ Transitions CSS pour les animations

CONCLUSION

Félicitations ! Vous venez de terminer 12 exercices d'intégration HTML, CSS et JavaScript !

Ce que vous avez appris :

✅ PARTIE 3 - Formulaires et Validation

- Formulaire de contact avec validation en temps réel
- Formulaire d'inscription avec indicateur de force du mot de passe
- Validation côté client avec JavaScript
- Messages d'erreur personnalisés

✅ PARTIE 4 - Projets Avancés

- Accordéon FAQ interactif
- Fenêtre modale (popup)
- Carrousel d'images
- Galerie avec filtres dynamiques

Continuez à pratiquer et créez vos propres projets !