

# Cours sur le RGAA (Référentiel Général d'Amélioration de l'Accessibilité)

---

## Qu'est-ce que le RGAA ?

Le **RGAA** est un référentiel français qui a pour but de **rendre les services numériques accessibles à tous**, en particulier aux personnes en situation de handicap.

Il traduit les exigences du standard international **WCAG 2.1** (Web Content Accessibility Guidelines) en critères concrets et applicables pour les sites web français.

---

## Objectifs du RGAA

- Permettre l'accès à l'information et aux services pour tous.
  - Réduire les barrières numériques (déficience visuelle, auditive, motrice, cognitive...).
  - Respecter la **loi française** sur l'accessibilité numérique (obligatoire pour les services publics).
- 

## Cadre légal

- **Obligation** pour :
    - Les sites publics (administrations, collectivités, universités...).
    - Certains sites privés (grandes entreprises ou prestataires de services publics).
  - **Décret n° 2019-768** du 24 juillet 2019.
  - Obligation de :
    - Réaliser un **audit d'accessibilité**.
    - Publier une **déclaration d'accessibilité**.
    - Mettre en place un **schéma pluriannuel et un plan d'action annuel**.
- 

## Structure du RGAA

Le RGAA se base sur :

- **106 critères de conformité** (issus des WCAG 2.1 niveau AA).
- Organisés en **thématiques** :
  1. Images
  2. Cadres (iframes)
  3. Couleurs
  4. Multimédia
  5. Tableaux
  6. Liens
  7. Scripts
  8. Éléments obligatoires
  9. Structure
  10. Présentation de l'information

- 11. Formulaires
  - 12. Navigation
  - 13. Consultation
  - 14. Documents bureautiques en ligne
- 

## Bonnes pratiques pour les développeurs

### 1. Images

- Ajouter des **attributs alt** pertinents aux images.
- Ne pas utiliser d'image comme seul moyen de transmettre une information.

### 2. Liens

- Les liens doivent être **clairs et explicites**.
- Éviter les liens "cliquez ici".

### 3. Contrastes

- Respecter un **contraste suffisant** entre le texte et l'arrière-plan (au moins **4.5:1**).

### 4. Navigation clavier

- Tout doit être accessible au **clavier uniquement** (sans souris).
- Utiliser des balises sémantiques : **button**, **nav**, **header**, **main**, **footer**...

### 5. Formulaires

- Associer chaque **input** à un **label**.
- Utiliser des messages d'erreur **clairs et accessibles**.
- Gérer le focus et les indications de champ obligatoire.

### 6. Multimédia

- Fournir des **sous-titres** pour les vidéos.
  - Fournir une **transcription texte** si besoin.
- 

## Exemple simple : champ de formulaire accessible

```
<form>
  <label for="email">Adresse e-mail :</label>
  <input type="email" id="email" name="email" required>
</form>
```

## Exemple de test manuel RGAA

Critère 6.1 : Chaque lien est-il explicite ?

- ☒ Oui : `<a href="/contact">Accéder au formulaire de contact</a>`
- ☐ Non : `<a href="/contact">Cliquez ici</a>`

Critère 3.3 : Le contraste entre le texte et le fond est-il suffisant ?

- Utiliser un outil comme [WebAIM Contrast Checker](#)
- Exemple : du texte gris clair (#aaa) sur fond blanc ne passe **pas** le test.

Critère 11.1 : Chaque champ de formulaire est-il correctement étiqueté ?

- ☒ Oui :

```
<label for="nom">Nom :</label>
<input type="text" id="nom" name="nom">
```

- ☐ Non :

```
<input type="text" placeholder="nom">
```

## Outils utiles pour l'accessibilité

Voici une sélection d'outils que les développeurs peuvent utiliser pour **tester et améliorer l'accessibilité** de leurs sites web :

- **Wave – WebAIM** : outil en ligne qui affiche visuellement les erreurs d'accessibilité directement sur votre page.
- **axe DevTools (Chrome/Firefox)** : plugin qui s'intègre dans les DevTools pour analyser les problèmes d'accessibilité automatiquement.
- **Color Contrast Checker (WebAIM)** : permet de vérifier les contrastes entre texte et arrière-plan selon les niveaux AA et AAA.
- **Lecteurs d'écran** :
  - **NVDA** (Windows, gratuit)
  - **VoiceOver** (macOS et iOS, intégré)
  - **JAWS** (Windows, payant)

---

## ☒ À faire en entreprise ou sur un projet

Voici une checklist simplifiée pour appliquer le RGAA efficacement dans un projet web :

- ☐ Suivre les **critères RGAA** dès la conception (UX/UI inclus).
- ☐ Utiliser des composants accessibles (boutons, menus, formulaires...).
- ☐ Intégrer les **tests d'accessibilité** dans les revues de code.
- ☐ Rédiger une **déclaration d'accessibilité** sur le site.
- ☐ Mettre en place un **schéma pluriannuel** et un **plan d'action annuel**.
- ☐ Effectuer des **audits réguliers**, manuels et automatiques.

- ☐ Sensibiliser et former les équipes techniques et fonctionnelles.
- 

## Ressources officielles

Pour aller plus loin et rester conforme à la réglementation :

- **Site officiel du RGAA** : <https://accessibilite.numerique.gouv.fr>
  - **Liste complète des critères RGAA 4.1** : <https://accessibilite.numerique.gouv.fr/methode/criteres/>
  - **Référentiel WCAG 2.1 (W3C)** : <https://www.w3.org/TR/WCAG21/>
  - **Guide développeur Access42** : <https://access42.net>
- 

## Conclusion

L'accessibilité numérique n'est **ni un luxe ni une option** : c'est une exigence légale, éthique et technique.

En suivant les critères du RGAA, tu :

- Respectes la loi française.
- Permetts l'accès aux contenus à tous les utilisateurs.
- Améliores la qualité globale et la maintenabilité de tes interfaces.

 **"Le web est pour tout le monde, sans exception." – Tim Berners-Lee**