

MANUEL MJML

Guide pratique en français

MJML - Mailjet Markup Language

Création d'emails responsives - Mac - HTML - JSON
Version V001 - 8 octobre 2026

Nom canonique : MANUEL_MJML_FR_2026-10-08_V001.docx

Référence. Guide pédagogique en français fondé sur la documentation officielle MJML consultée le 8 octobre 2026. Ce document n'est pas une traduction officielle de l'équipe MJML.

Sommaire

1. Comprendre MJML
2. Démarrer avec l'éditeur en ligne
3. Structure minimale
4. Sections et colonnes
5. Texte et typographie
6. Images
7. Boutons, liens et séparateurs
8. Styles globaux et classes
9. Responsive et mobile
10. Validation et erreurs
11. Conversion MJML vers HTML
12. Installation sur macOS
13. Ligne de commande
14. MJML au format JSON
15. JSON MJML vers HTML
16. Workflow conseillé
17. Dépannage
18. Aide-mémoire
- Annexe A. Modèle complet
- Annexe B. Exemple JSON
- Sources officielles

1. Comprendre MJML

MJML - Mailjet Markup Language - est un langage de balisage conçu pour simplifier la création d'emails HTML responsives. Au lieu d'écrire directement les tables HTML et les règles spécifiques à chaque logiciel de messagerie, vous décrivez la structure avec des composants tels que `mj-section`, `mj-column`, `mj-text` et `mj-image`. Le moteur MJML génère ensuite le HTML final.

À retenir. Le fichier `.mjml` est un fichier source. Le destinataire reçoit du HTML - HyperText Markup Language - compilé à partir du MJML, pas le MJML lui-même.

```
<mjml>
  <mj-body>
    <mj-section>
      <mj-column>
        <mj-text>Bonjour</mj-text>
      </mj-column>
    </mj-section>
  </mj-body>
</mjml>
```

- Syntaxe plus lisible que le HTML d'email classique.
- Composants responsives prévus pour ordinateur et mobile.
- Génération automatique de nombreux correctifs pour Outlook et autres clients.
- Possibilité de travailler en balisage MJML ou sous forme d'objet JSON - JavaScript Object Notation.

2. Démarrer avec l'éditeur en ligne

Pour débiter sans installer de logiciel, utilisez l'éditeur officiel MJML en ligne. Il permet de coller du MJML et de voir immédiatement le rendu compilé.

Éditeur officiel : <https://mjml.io/try-it-live>

1. Ouvrir votre fichier `.mjml` avec un éditeur de texte.
2. Copier tout le code.
3. Ouvrir l'éditeur MJML en ligne.
4. Remplacer le code d'exemple par votre code.
5. Observer la prévisualisation.
6. Corriger les erreurs signalées.
7. Exporter ou copier le HTML compilé.

Images. Pour un email réellement envoyé, les images doivent normalement être disponibles via une adresse HTTPS publique. Un chemin local comme `assets/photo.png` n'est pas accessible au destinataire.

3. Structure minimale

Un document MJML commence par la balise racine `mjml`. Il peut contenir `mj-head` pour les réglages globaux et contient `mj-body` pour le contenu visible.

```
<mjml lang="fr">
  <mj-head>
    <!-- styles et paramètres -->
  </mj-head>
  <mj-body width="600px">
    <mj-section>
      <mj-column>
        <mj-text>Mon contenu</mj-text>
      </mj-column>
    </mj-section>
  </mj-body>
```

</mjml>

Élément	Fonction
mjml	Racine du document.
mj-head	Styles, classes, polices et paramètres globaux.
mj-body	Contenu visible de l'email.
mj-section	Ligne horizontale de mise en page.
mj-column	Colonne responsive contenant les composants.

Largeur. La documentation MJML indique que l'espace d'une section est de 600 px par défaut, valeur modifiable avec l'attribut width de mj-body.

4. Sections et colonnes

Les sections organisent les lignes de l'email. À l'intérieur de chaque section, les colonnes répartissent la largeur disponible. Les colonnes se réorganisent verticalement sur mobile selon le comportement responsive de MJML.

```
<mj-section>
  <mj-column width="50%">
    <mj-text>Gauche</mj-text>
  </mj-column>
  <mj-column width="50%">
    <mj-text>Droite</mj-text>
  </mj-column>
</mj-section>
```

- Une section doit contenir des colonnes, même lorsqu'il n'y en a qu'une.
- La somme des largeurs des colonnes ne doit pas dépasser la largeur du parent.
- Une colonne ne doit pas contenir une autre mj-column ou une mj-section.
- Chaque composant placé dans une colonne occupe la largeur disponible de cette colonne, sauf réglage spécifique.

5. Texte et typographie

mj-text contrôle le texte. Les attributs usuels sont font-family, font-size, line-height, color, align, padding et font-weight.

```
<mj-text
  font-family="Futura, Trebuchet MS, Arial, sans-serif"
  font-size="20px"
  line-height="28px"
  color="#201820"
  align="center">
  Visages pour le Futur
</mj-text>
```

Polices. Une police installée sur votre Mac n'est pas automatiquement disponible chez le destinataire. Prévoyez toujours plusieurs polices de remplacement.

Usage	Valeur pratique
Texte courant	15 à 18 px
Sous-titre	18 à 24 px
Titre	28 à 36 px selon la longueur
Interligne	Environ 1,35 à 1,6 fois la taille du texte

6. Images

mj-image insère une image. Utilisez src pour l'adresse, alt pour le texte alternatif, width pour la largeur et href si l'image doit être cliquable.

```
<mj-image
  src="https://exemple.org/images/blason_thonex.png"
  alt="Armoiries de la commune de Thônex"
  width="82px"
/>
```

Chemin	Prévisualisation locale	Email envoyé
assets/image.png	Possible selon l'outil	Non fiable
file:///Users/.../image.png	Seulement sur votre Mac	À proscrire
https://site.ch/image.png	Oui	Oui

Accessibilité. Renseignez toujours alt avec une description utile quand l'image transmet une information. Pour une image strictement décorative, l'alt peut être vide selon votre stratégie d'accessibilité.

7. Boutons, liens et séparateurs

Bouton

```
<mj-button
  href="https://exemple.org/projet"
  background-color="#0078B9"
  color="#ffffff">
  Découvrir le projet
</mj-button>
```

Lien dans un texte

```
<mj-text>
  <a href="https://exemple.org">En savoir plus</a>
</mj-text>
```

Séparateur et espace

```
<mj-divider border-color="#D02028" border-width="1px" />
<mj-spacer height="24px" />
```

8. Styles globaux et classes

Dans mj-head, mj-attributes permet de définir des valeurs par défaut. mj-all affecte tous les composants, tandis que mj-class crée un groupe réutilisable d'attributs.

```
<mj-head>
  <mj-attributes>
    <mj-all font-family="Arial, sans-serif" />
    <mj-text padding="0" />
    <mj-class name="titre" color="#0078B9" font-size="30px" />
  </mj-attributes>
</mj-head>
<mj-text mj-class="titre">Visages pour le Futur</mj-text>
```

Avantage. Centraliser les styles limite les répétitions et facilite une modification globale de la newsletter.

9. Responsive et mobile

MJML est conçu pour produire des emails responsives. Les colonnes d'une section peuvent se juxtaposer sur ordinateur et s'empiler sur un petit écran. Le HTML généré inclut notamment des media queries et des adaptations pour certains clients de messagerie.

- Conservez une largeur générale raisonnable, souvent 600 px.
- Évitez des textes trop petits.
- Ne construisez pas une mise en page dont le sens dépend uniquement d'un affichage côte à côte.
- Testez le rendu sur plusieurs largeurs et plusieurs clients de messagerie.
- N'utilisez pas uniquement l'aperçu d'un navigateur comme preuve de compatibilité email.

10. Validation et erreurs

MJML propose plusieurs niveaux de validation. La ligne de commande accepte notamment strict, soft et skip. Le niveau soft est le comportement par défaut indiqué dans la documentation actuelle.

```
mjml input.mjml -o output.html --config.validationLevel strict
```

Niveau	Comportement
strict	Les erreurs de validation provoquent un échec explicite.
soft	Le HTML peut être généré avec une liste d'erreurs.
skip	Validation ignorée.

- Vérifiez les balises fermantes.
- Vérifiez que mj-column est placé dans mj-section.
- Vérifiez les unités px ou %.
- Vérifiez les URL d'images.
- Vérifiez les caractères spéciaux et l'encodage UTF-8.

11. Conversion MJML vers HTML

La conversion transforme la source MJML en HTML de courriel. En ligne, l'éditeur officiel réalise cette compilation automatiquement. En local, la commande officielle est simple :

```
mjml input.mjml -o output.html
```

Le fichier output.html peut ensuite être ouvert dans Safari, Firefox ou Chrome pour un contrôle visuel général, puis testé dans de vrais logiciels de messagerie avant diffusion.

12. Installation sur macOS

MJML s'installe avec npm - Node Package Manager. La documentation officielle indique la procédure locale suivante :

```
mkdir mon-email
cd mon-email
npm init -y
npm install mjml
```

Vous pouvez ensuite lancer le binaire installé localement :

```
./node_modules/.bin/mjml index.mjml -o index.html
```

Pré-requis. npm est fourni avec Node.js. Installez une version maintenue de Node.js avant MJML si npm n'est pas déjà disponible sur votre Mac.

13. Ligne de commande

Commande	Effet
<code>mjml index.mjml -o index.html</code>	Compile vers un fichier HTML.
<code>mjml -r index.mjml</code>	Affiche le HTML compilé dans le Terminal.
<code>mjml --watch index.mjml -o index.html</code>	Surveille le fichier et recompilé automatiquement à chaque modification.
<code>mjml input.mjml --config.validationLevel strict</code>	Compile avec validation stricte.

Si vous ajoutez le dossier `./node_modules/.bin` à votre PATH, vous pouvez appeler directement `mjml` au lieu du chemin complet.

14. MJML au format JSON

MJML peut aussi être représenté comme un objet JSON - JavaScript Object Notation. D'après la documentation officielle, chaque composant possède un `tagName`, un objet `attributes` et soit `content`, soit un tableau `children`.

```
{
  "tagName": "mj-text",
  "attributes": {
    "font-size": "20px",
    "color": "#0078B9"
  },
  "content": "Visages pour le Futur"
}
```

Usage. Le JSON est surtout utile pour générer ou modifier programmatiquement des emails. Un visualiseur JSON affiche la structure des données, pas le rendu graphique final de l'email.

15. JSON MJML vers HTML

La fonction `mjml2html` du paquet MJML accepte directement la définition JSON native MJML.

```
import fs from "node:fs";
import mjml2html from "mjml";

const json = JSON.parse(fs.readFileSync("email.json", "utf8"));
const result = await mjml2html(json);
fs.writeFileSync("email.html", result.html, "utf8");
```

Il faut distinguer un JSON MJML natif d'un JSON produit par un convertisseur XML vers JSON générique. Ce dernier n'utilise pas nécessairement `tagName`, `attributes`, `children` et `content` dans la structure attendue par MJML.

16. Workflow conseillé

1. Créer ou reconstruire la maquette en MJML.
2. Prévisualiser dans l'éditeur MJML.
3. Séparer les textes, images et liens.
4. Héberger les images finales en HTTPS.
5. Valider la structure MJML.
6. Compiler en HTML.
7. Ouvrir le HTML dans un navigateur pour contrôle général.

8. Tester le HTML dans les principaux clients de messagerie.
9. Conserver le .mjml comme source maître.
10. N'utiliser JSON que si une automatisation ou une manipulation programmée le justifie.

17. Dépannage

Problème	Cause probable	Correction
L'image ne s'affiche pas	Chemin local	Utiliser une URL HTTPS publique.
Les colonnes sont incorrectes	Structure ou largeurs incohérentes	Vérifier section, colonnes et pourcentages.
La police change	Police absente chez le destinataire	Ajouter des polices de remplacement.
Le navigateur affiche du code au lieu de l'email	HTML enregistré comme texte échappé	Réenregistrer comme vrai HTML UTF-8.
Le JSON ne produit rien	JSON non natif MJML	Vérifier tagName, attributes, children/content.
Outlook diffère du navigateur	Moteur de rendu différent	Tester le HTML compilé dans Outlook et simplifier les éléments fragiles.

TextEdit sur Mac. TextEdit peut transformer du code HTML en document HTML affichant le code comme texte si le fichier est enregistré dans le mauvais mode. Pour éditer du code, utilisez du texte brut et évitez toute conversion automatique en texte enrichi.

18. Aide-mémoire

Besoin	Composant / commande
Créer une ligne	<code>mj-section</code>
Créer une colonne	<code>mj-column</code>
Ajouter du texte	<code>mj-text</code>
Ajouter une image	<code>mj-image</code>
Ajouter un bouton	<code>mj-button</code>
Ajouter une séparation	<code>mj-divider</code>
Créer un espace vertical	<code>mj-spacer</code>
Définir des styles globaux	<code>mj-attributes / mj-class</code>
Compiler	<code>mjml input.mjml -o output.html</code>
Validation stricte	<code>--config.validationLevel strict</code>
Surveillance automatique	<code>--watch</code>
Manipulation programmée	JSON natif MJML + <code>mjml2html</code>

Annexe A. Modèle complet

```
<mjml lang="fr">
  <mj-head>
```

```

<mj-preview>Visages pour le Futur</mj-preview>
<mj-attributes>
  <mj-all font-family="Arial, sans-serif" />
  <mj-text color="#201820" font-size="16px" line-height="24px" />
  <mj-class name="titre" color="#0078B9" font-size="30px" font-weight="700" />
</mj-attributes>
</mj-head>

<mj-body width="600px" background-color="#ffffff">
  <mj-section padding="32px 20px">
    <mj-column>
      <mj-image
        width="82px"
        src="https://exemple.org/blason_thonex.png"
        alt="Armoiries de Thônex"
      />
      <mj-text mj-class="titre" align="center">
        Visages pour le Futur
      </mj-text>
      <mj-text align="center">
        Une mission photographique liée au territoire des Trois-Chêne.
      </mj-text>
      <mj-button href="https://exemple.org/projet">
        Découvrir le projet
      </mj-button>
    </mj-column>
  </mj-section>
</mj-body>
</mjml>

```

Annexe B. Exemple JSON natif MJML

```

{
  "tagName": "mjml",
  "attributes": {"lang": "fr"},
  "children": [
    {
      "tagName": "mj-body",
      "attributes": {"width": "600px"},
      "children": [
        {
          "tagName": "mj-section",
          "attributes": {},
          "children": [
            {
              "tagName": "mj-column",
              "attributes": {},
              "children": [
                {
                  "tagName": "mj-text",
                  "attributes": {"align": "center"},
                  "content": "Visages pour le Futur"
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

Sources officielles

Sources consultées le 8 octobre 2026 :

- MJML Guide - documentation officielle: <https://documentation.mjml.io/>
- MJML - installation et ligne de commande: <https://mjml.io/download>
- MJML - éditeur en ligne: <https://mjml.io/try-it-live>
- Projet open source MJML sur GitHub: <https://github.com/mjmlio/mjml>

La documentation officielle reste la référence prioritaire pour les attributs précis de chaque composant et les changements de version. Les clients de messagerie évoluent ; un test réel avant diffusion reste nécessaire.