

# ■ Comparatif : register.js (ancien vs corrigé)

## 1 ■■■ Structure globale

- L'ancien code utilisait plusieurs variables booléennes (`let firstname = false; ...`) pour suivre les champs du formulaire. La nouvelle version les regroupe dans un objet `fields = { firstname: false, ... }`, plus propre et évolutif.

## 2 ■■■ Vérification des champs

L'ancien code utilisait des fonctions séparées (`checkFirstname`, `checkLastname`, etc.) et des écouteurs individuels. La nouvelle version lie directement les événements aux propriétés de `fields`, réduisant la duplication.

Exemple :

```
Ancien :  
function checkFirstname() { firstname = this.value.trim().length >= 2; checkAll(); }  
Nouveau :  
formEls.firstname.addEventListener('input', () => { fields.firstname =  
formEls.firstname.value.trim().length >= 2; checkAll(); });
```

## 3 ■■■ Validation de l'email

Même logique, mais le champ est désormais suivi via `fields.email` pour plus de cohérence et moins d'erreurs.

## 4 ■■■ Entropie et mot de passe

- L'ancien code calculait bien l'entropie mais ne mettait pas à jour la barre visuelle. La nouvelle version met à jour la couleur et la largeur dynamiquement selon la force détectée.

```
Ancien :  
let width = 20; let color = 'red'; // jamais modifiés  
Nouveau :  
switch(entropy) { case 'STRONG': color='lightgreen'; width=80; fields.password=true;  
break; }
```

## 5 ■■■ Fonction checkAll()

L'ancien code vérifiait chaque variable individuellement. La nouvelle version utilise `Object.values(fields).every(Boolean)` pour activer le bouton dès que tous les champs sont valides.

## 6 ■■■ Correction du HTML

Un input manuel doublonnait un champ généré par Symfony, provoquant un bug dans la sélection par ID. Ce champ a été supprimé.

## ■ Résumé global

Catégorie	Ancien code	Nouveau code
-----------	-------------	--------------

Structure	Variables séparées	Objet fields{} centralisé
Mot de passe	Couleurs statiques	Couleurs et largeur dynamiques
Validation	Bouton jamais activé	Bouton activé quand tout est bon
Lisibilité	Fonctions dispersées	Code compact et clair

■ Résultat : la barre de force du mot de passe est fonctionnelle, la validation est réactive, et le code est maintenable.