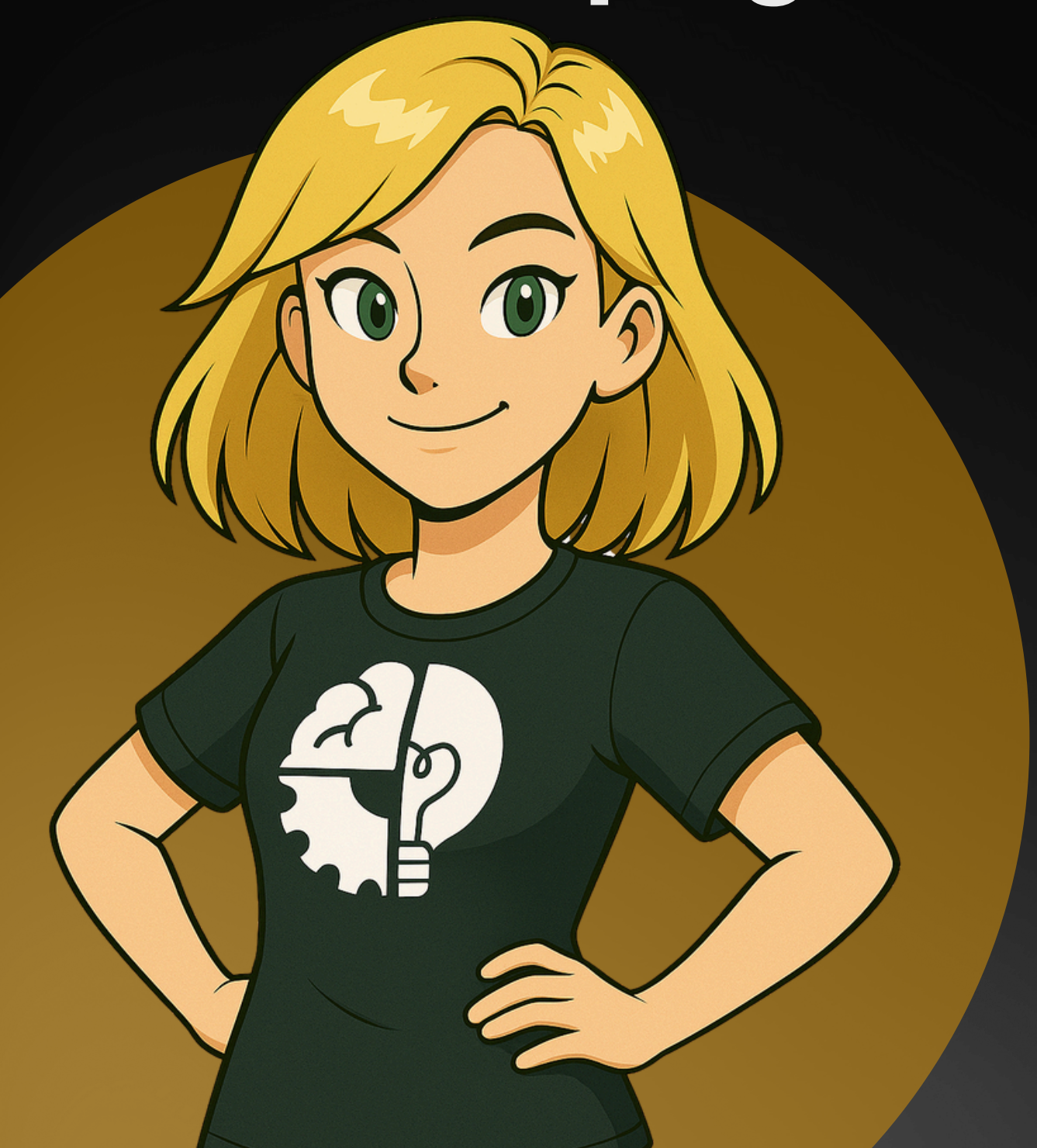
90 % DES DEVS ANGULAR UTILISENT MAL MAP

Es-tu sûr de ne pas tomber dans le piège ?



Elodie TURLIER

Angular, IA, agilité & storytelling : construire des apps prêtes à survivre à l’apocalypse digitale | Du besoin à la livraison

MAP N'EST PAS CE QUE TU CROIS

01

map n'est pas juste là pour transformer des valeurs.

C'est une **fonction de projection** :

→ Transformation déclarative

→ Zéro mutation

→ Aucun effet de bord

```
of(1, 2, 3).pipe(
  map(n => n * n)
).subscribe(console.log); // 1, 4, 9
```

⚠ **À savoir :**

map fonctionne **uniquement sur des transformations synchrones**.

Si *n * n* devait être une promesse ou une requête API, map ne suffirait pas.

Dans ce cas → utilise *switchMap*, *mergeMap*, etc.

Elodie TURLIER
Angular, IA, agilité & storytelling

ADAPTER TES DONNÉES AVEC MAP : L'ARME SECRÈTE

02

Avec Angular, les API retournent des Observables.

Pour rendre les données vraiment utilisables, tu dois les adapter au bon format.

Exemple : convertir une chaîne en *Date* :

```
this.http.get<Article[]>('/api/articles').pipe(
  map(articles => articles.map(article => ({
    ...article,
    datePublication: new Date(article.datePublication)
  })))
).subscribe(articles => this.articles = articles);
```

✅ Typage respecté
✅ Moins de bugs
✅ Données prêtes à l'emploi

Elodie TURLIER
Angular, IA, agilité & storytelling

LES 3 ERREURS QUI RUINENT TON CODE AVEC MAP

03

❌ **Erreur #1 :** Typage flou, any partout
→ typage strict sur l'Observable.
⚠ Sans typage, Angular ne détecte pas les incohérences.

❌ **Erreur #2** : Calculs lourds directement dans map
→ Préfère la mémoization ou déporte les calculs ailleurs.
⚠ Une fonction complexe dans map va ralentir ton flux.

❌ **Erreur #3** : Effets de bord dans map (logs, mutations)
→ Pour ça, utilise tap, jamais map.
⚠ Un simple console.log() dans map = piège courant !

Elodie TURLIER
Angular, IA, agilité & storytelling

LES 4 RÈGLES D'OR POUR UN MAP EFFICACE.

04

✓ map = **pure function** : pas d'effet de bord

✓ Ne jamais muter les objets : utilise la déstructuration

✓ Typage strict des réponses HTTP :

```
this.http.get<{resultats: Article[]}>('/api/recherche')
    .pipe(
        map(response => response.resultats)
    );
```

✓ Toujours préférer pipe(map(...))

Elodie TURLIER
Angular, IA, agilité & storytelling

2 COMBOS IMPARABLES AVEC MAP

05

💡 map + filter : transformer puis filtrer

typescript

```typescript
this.form.get('email').valueChanges.pipe(
  map(email => email.trim().toLowerCase()),
  filter(email => email.includes('@'))
);
```

💡 map + switchMap : transformer puis enchaîner une requête dépendante

```typescript
this.route.paramMap.pipe(
  map(params => +params.get('id')),
  switchMap(id =>
this.service.fetchDetails(id))
);
```

Elodie TURLIER
Angular, IA, agilité & storytelling

COMBOS AVANCÉS :06 QUAND MAP NE SUFFIT PLUS.

🔹 **map + switchMap** = transformation + requête dépendante

🔹 **map + mergeMap** = transformation + parallélisme (plusieurs Observables en même temps)

🔹 **map + scan** = transformation + accumulation (compteur, somme, historique...)

💡 **Astuce** : Le bon combo dépend de ton besoin : Synchrone, asynchrone ou accumulation ?
Choisis l’opérateur adapté.

Elodie TURLIER
Angular, IA, agilité & storytelling

MAP NE PROTÈGE PAS TON PIPELINE. VOICI COMMENT FAIRE.

07

map transforme tes données.

Mais il ne gère pas les erreurs.

➡ Utilise catchError juste après map :

```
this.http.get('/api/data').pipe(
  map(data => data.entries),
  catchError(err => {
    console.error('Échec de transformation:', err);
    return of([]);
  })
);
```

✅ Protection solide, même après transformation.

Elodie TURLIER
Angular, IA, agilité & storytelling

MAP N’EST PAS TOUJOURS LA SOLUTION : VOICI QUAND L’ÉVITER.

08

⚠ **Évite map si :**

- Tu dois accumuler des valeurs → utilise scan.
- Tu travailles avec des opérations asynchrones → préfère switchMap, mergeMap.
- Tu veux faire des effets de bord → utilise tap.

Règle simple :

map = projection pure.

Pas accumulation. Pas asynchrone. Pas action.

Elodie TURLIER
Angular, IA, agilité & storytelling

MAP : LES 4 CLÉS À NE JAMAIS OUBLIER.

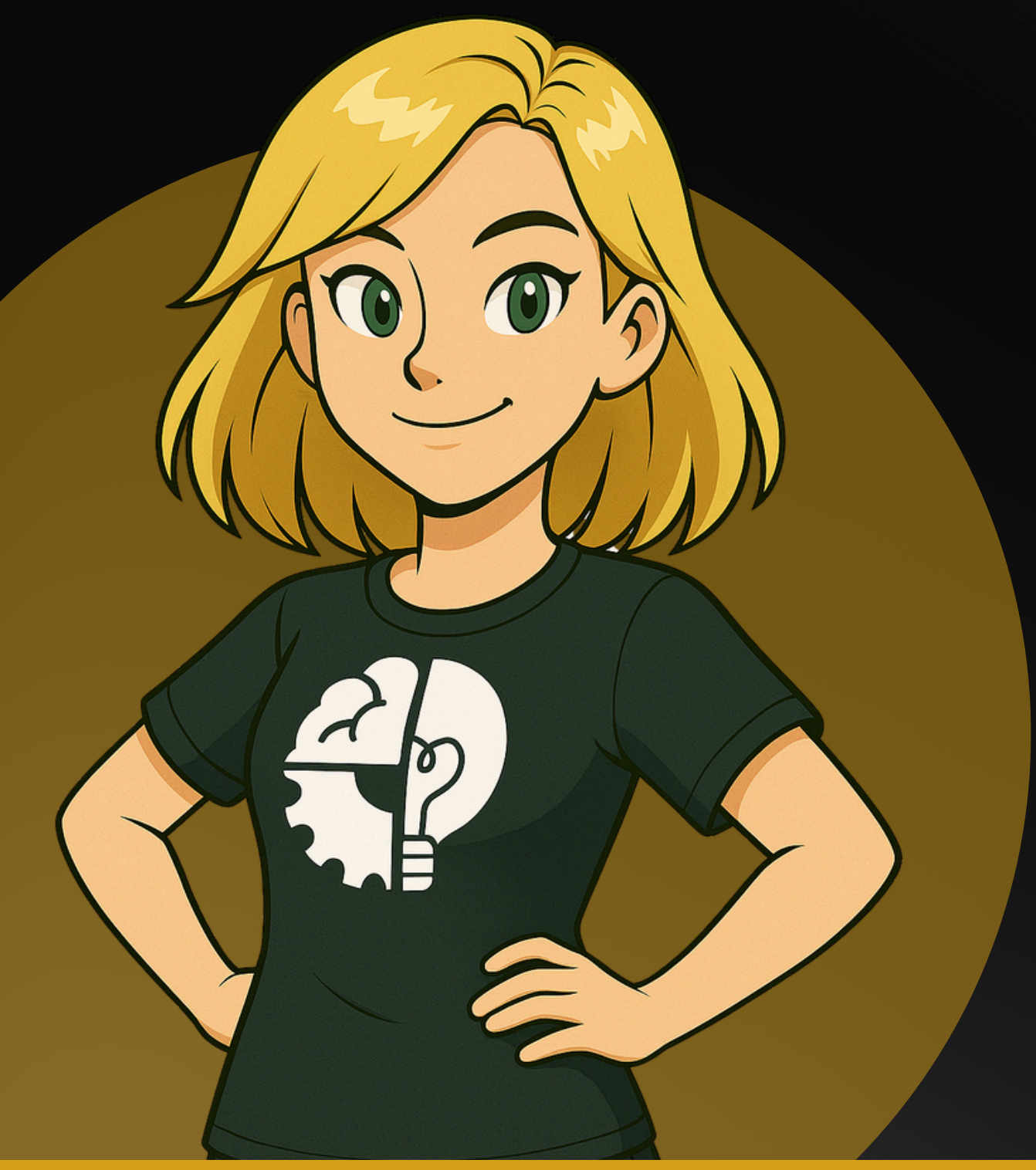
✅ Jamais d’effets de bord dans map
✅ Toujours typer strictement tes données
✅ Préfère la syntaxe pipe(map(...))
✅ Combine-le intelligemment avec filter, switchMap, etc.

✔️ Si tu respectes ces règles :
→ Moins de bugs
→ Code plus clair
→ Maintenance facilitée

Pour en savoir plus:
https://rxjs.dev/api/operators/map

Elodie TURLIER
Angular, IA, agilité & storytelling

ABONNE TOI POUR PLUS DE CONTENU



Elodie TURLIER

Angular, IA, agilité & storytelling : construire des apps prêtes à survivre à l’apocalypse digitale | Du besoin à la livraison

Enregistre ou partage si ça t’a été utile