TAKE AWAY

LES RESULT / EITHER

A utiliser pour modéliser les cas d’erreur

Sécurisant

Facile à manipuler

Parfois encodé avec un seul paramètre, pour homogénéiser les erreurs, s’appelle souvent Try dans ce cas.

```
interface MyDomainErrors extends Error { };
type Try<A> = Either<MyDomainErrors, A>;
```

MORE

ACCUMULATEURS D'ERREURS

On souhaite parfois remonter l'ensemble des erreurs rencontrées (formulaire, json, csv, compilation, …) : nous avons besoin d'accumulateurs d'erreurs **!**

Problème Either **va contenir la première erreur qui sera propagée par** chain **ou** map

On aimerait quelque chose qui ressemble **à**

```
type AccumulateErrors<E,A> = Either<NonEmptyArray<E>, A>;
```

… Ce type s'appelle parfois VALIDATION