

U0LPMPL2U : under the hood, `Json.Decode` may have some fancy stuff but you never need to deal with that
 U6D1E91QV : i'm specifically looking for that fancy stuff though
 U6D1E91QV : i am trying to construct a record piece by piece from results
 U6D1E91QV : from a form record
 U0LPMPL2U : You can do that in pure elm using the functions provided by `Json.Decode`
 U6D1E91QV : there's no json involved there, i am merely trying to copy the api
 U0LPMPL2U : Ah, so you have many `Result` values that you want to combine together?
 U6D1E91QV : in a way -- and collect in a record
 U6D1E91QV : can i paste stuff in here?
 U0LPMPL2U : sure that would be helpful :slightly_smiling_face:
 U6D1E91QV : ```type FieldValue raw val err
 = Empty
 | Partial raw
 | Invalid raw err
 | Valid val
 ...

U6D1E91QV : that's a field type for a for field, which gets updated upon input and may or may not be valid
 U0LPMPL2U : OK
 U6D1E91QV : ```extractField : FieldValue raw val err -> Result (List err2) val
 extractField =
 value &&& Result.fromMaybe []
 ...

U6D1E91QV : this will turn any FieldValue into either a `val` or a list of errors
 U6D1E91QV : example use```
 extractUsername form username = (extractField &&& .username)
 ...

U6D1E91QV : given some sort of form record with a field `username`, this will extract said value as a result
 U0LPMPL2U : Where does the list of errors come from? Doesn't the `Invalid` tag only allow one error?
 U6D1E91QV : (excuse the typo please)
 U6D1E91QV : the list is always empty initially
 U6D1E91QV : the errors collected are not on a field, but on a form level
 U6D1E91QV : i mean to add additional validators that validate on a form level, e.g. can check if password1 and password2 are the same
 U0LPMPL2U : hmmm... since `extractField` only deals with a single field, shouldn't it only return a single error?