U23SA861Y : Dict
U23SA861Y : Task
U0LPMPL2U : `Maybe.Extra` has a function `unwrap : b -&gt; (a -&gt; b) -&gt; Maybe a -&gt; b`
U0LPMPL2U : So you could say `Maybe.Extra.unwrap (text "") Calendar.view model.calendar`
U62JFJWKT : I'm friendly with map ruby is my defaut langage
U4872964V : <@U0LPMPL2U> functions with three or more parameters are harder to understand though
U0LPMPL2U : fair
U4872964V : but it's a matter of taste i suppose
U4872964V : i like pipelining
U0LPMPL2U : I'd say `case` vs `map |&gt; withDefault` vs `unwrap` are probably context dependent for readability
U6478N1V1 : Hi, could some one help me explain why these 2 expressions yield different results:
```
import Html exposing (text)

x1 = (*) ((*) 2 3) 3 ^ (^) 2 2 + 5 --results in 104981
x2 = (*) 2 3 * 3 ^ (^) 2 2 + 5 --results in 491

main = text (toString (x1 == x2))
```

x2 only replaces infix operations with prefix operators of x1

U0CLDU8UB : What I like to do with views is have a `maybeToList : Maybe a -&gt; List a`This lets me do compose views like so:
```
view model =
    div []
        ( maybeToList model.maybeOne ++ maybeToList model.maybeTwo ++ [ someView, otherView ] )
```

U3LUC6SNS : I believe that it is the case that languages/os's designed by one person, maybe two (see: Unix) are more successful than those designed by a large group. There needs to a single coherent vision. But I'd love to see both examples and counterexamples.
U4872964V : <@U0CLDU8UB> yes, that's useful
U0CLDU8UB : nbergsma: I think `(*) ((*) 2 3) 3` is not equal to `(*) 2 3 * 3`
U4872964V : it's in maybe-extra too
U0CLDU8UB : yep, I rarely have many cases for *.Extra though
U0CLDU8UB : See, `(*) 2 3` has function precedence, which is the highest precedence in Elm.
U6478N1V1 : Hi <@U0CLDU8UB>, could you explain?
U6478N1V1 : ah equal to 0 right?
U0CLDU8UB : Oh wait, that can't be the source, product is always the same no matter the order