```
U153UK3FA: Changing `expect = Http.expectStringResponse (\_-> Ok ())` to be `, expect =
Http.expectStringResponse (\_ -> Ok "pizza")` will make the result always be "pizza" ignoring the actual response
U4H406NET: Thank you, it compiled and matched the error case in the update function.
U68471J6P: I'm thinking of building my first "real" Elm website with Elixir for the back-end. Which part should I build
first, the front-end or the back-end?
U0CLDU8UB: I'd say either way you will be building them side by side
U0CLDU8UB: Building the frontend makes you realize the API needs to be changed, so then you switch to the
U5P4FLYLE: Hi, I have the following compilation error: "The definition of 'rangeFrameHintDot' does not match its type
annotation.
100| rangeFrameHintDot: Maybe Point -> Float -> Float -> DataPoint Msg
101| rangeFrameHintDot hinted x y =
102|\> \{ view = Just (myCircle x y) \}
103|> , viewHint = Nothing
104|\> , x = x
105|> , xLine = Maybe.andThen (flashyLine x y) hinted
106|\>, xTick = Just (simpleTick x)
107|\> , y = y
108|> , yLine = Maybe.andThen (flashyLine x y) hinted
109|> , yTick = Just (simpleTick y)
110|> }
The type annotation for `rangeFrameHintDot` says it always returns: DataPoint Msg
But the returned value (shown above) is a:
{ view : Maybe (Svg Msg)
, viewHint : Maybe a
, x : Float
, xLine: Maybe (AxisSummary -> LineCustomizations)
, xTick: Maybe TickCustomizations
, y: Float
, yLine: Maybe (AxisSummary -> LineCustomizations)
, yTick : Maybe TickCustomizations
Hint: The record fields do not match up. Maybe you made one of these typos?
                                                                                    x <-&gt; y
                                                                                                  xLine &lt:-&at:
vLine xTick <-&gt; vTick
                                 y <-&gt; x yLine &lt;-&gt; xLine
                                                                      vTick <-&at; xTick``
I look above and see the same thing...Do you know how to proceed in such a case?
U3SJEDR96: could you show the definition of `DataPoint`?
U3SJEDR96: oh, there <a href="http://package.elm-lang.org/packages/terezka/elm-plot/latest/Plot#DataPoint">http://package.elm-lang.org/packages/terezka/elm-plot/latest/Plot#DataPoint</a>
U3SJEDR96: you have 'viewHint' rather than 'hint'
U5P4FLYLE: <@U3SJEDR96> yes it should be `hint`. Thanks!!!
U666LA9NK: Hi, I am very new to elm. I have questions regarding good practices of updating internal intermediate
status. If I have the case of ""
type alias Model =
 { number : Int
 , numberIsTooLarge: Bool
;;
}
I want `numberIsTooLarge` to be automatically be updated along with `number`. What is the best practice to do this?
Shall I include the code to update `numberIsTooLarge` when handling all the cases that `number` is updated, or is there
a better way to do that in elm?
U2XRG0UKA: When does you use `numberIsTooLarge`? Do you reference it once in the view to display an error
U153UK3FA: don't store what you can calculate
U2FL3ARLG: right, I'd say you don't need `numberIsTooLarge` in your model, but a function `numberIsTooLarge:
Model -> Bool` that you can use anywhere
U666LA9NK: <@U2XRG0UKA> I use it in multiple places in view and when update some other messages.
U666LA9NK: <@U2FL3ARLG> veah vou are right. I did not think of that. Thanks~
U2SR9DL7Q: who likes a good math challenge?
U4872964V: who doesn't?: slightly_smiling_face:
```

U0FP80EKB: They're all good math challenges, Brent

U2SR9DL7Q: I'm trying to implement something in elm. A function is passed a positive integer, and returns a list of tuples that represent unique pairings of all non-negative integers up to and including the one passed.

U2SR9DL7Q: so `func 2` returns `[(0,0), (0, 1), (0,2), (1,1), (1,2), (2,2)]`

U57KYFW67: You'll probably want `concatMap` for that

U2SR9DL7Q: It would not return _both_ (1,2) and (2,1).

U57KYFW67: oh

U2SR9DL7Q: I'm been tinkering with concatMap....

U57KYFW67: Yeah, still. This should be possible with `concatMap` and then the equivalent of filtering. I know how

you'd do it in 4 lines of Haskell... but Elm doesn't have the same ease of working with lists-as-monads

U681TBBUP: Did you miss `(0,2)`??

U4872964V: <@U2SR9DL7Q> probably easiest to generate all and then filter out those that are in reverse order

U57KYFW67: Start by asking how you'd create the tuples for a fixed first element

U2SR9DL7Q : yes i did miss (0,2)

U0FP80EKB : Generate, then use something like `List.Extra.uniqueBy` where the function considers the flipped tuples the same

U681TBBUP: It really does seem like something you can accomplish via recursion

U57KYFW67: If you use the functions in `List`, you can avoid any explicit recursion. But maybe the recursive solution is clearer.

U2SR9DL7Q: I tried downloading the list-extra package, but it said it was incompatible with my current elm version.

I've just been re-implementing the funcs I need though from the source code.

U2SR9DL7Q: from the docs I thought `(List.range 0 highest) andThen \x -> (List.range 0 highest) andThen \y ->

[(x, y)] could give me the list of all the pairs, and then I would pass through it with `uniqueBy`

U2SR9DL7Q: But there's a syntax error in this somewhere...

U2SR9DL7Q: `andThen` from the list-extra package functioning as `flip concatMap`

U4872964V: you don't need `uniqueBy`, just remove all pairs that have the elements in the wrong order

U2SR9DL7Q: <@U4872964V> you think that would be easier?

U4872964V: `filter (\(x,y) -> x <= y)`

U3SJEDR96: it'll definitely be (much) faster, and doesn't require any -extra stuff

U681TBBUP: http://rosettacode.org/wiki/Combinations#Haskell

U2SR9DL7Q: Coolness. I still can't get the first function to work though. It's been awhile since I did these functional combinations.

U604S603Y: on dealing with Ports: Do I need to subscribe to them in Elm even when I'm only sending data to Javascript (in my case: setting the page title)?

U604S603Y: because I call the Port in Elm, I don't get an error but it never gets called in Javascript land

U4872964V: no you don't need to subscribe to the port to send to it (how would you do that, anyway)

U4872964V: you need to subscribe to it in javascript, though

U681TBBUP : Generating all permutations and filtering them is a bad idea because there are you are dealing with `n choose 2` vs `2^n`

U604S603Y: on closer inspection of my code and the signature - the Port emits a `Cmd` after being called, so I need to "forward" that to the Elm runtime again in my `update` function?

U3SJEDR96 : <@U604S603Y> can you show some code?

U4872964V: <@U681TBBUP>, it's only slightly less than double the elements, no biggie:slightly_smiling_face:

U4872964V: <@U604S603Y>, no, you return the `Cmd` from your update function

U2SR9DL7Q: <@U681TBBUP> <@U4872964V> fortunately, the highest number will likely never exceed 9

U604S603Y: yeah, I explicitly ignored the Cmd - no wonder Elm never did anything with it...

U604S603Y: thanks <@U4872964V> <@U3SJEDR96>

U681TBBUP: Ah well then it really doesn't matter what approach you take. If you wanted scalability though dynamic programming would be the way to go

U3SJEDR96 : <@U604S603Y> Elm has managed side-effects; which means that side-effects can only happen in the runtime. You create "commands" for the runtime to tell it to do something, but then of course, you have to hand them to the runtime for it to actually do something :slightly_smiling_face:

U48AEBJQ3 : <@U2SR9DL7Q> Are you just trying to write `List.map2 (List.map &It;&It; (,)) xs (tails xs) |> List.concat`?

U2SR9DL7Q: Okay, so we're almost there. I have ```

createDominoes: Int -> List Domino

createDominoes highest =

List.map (\x -> List.map (\y -> Domino x y <| List.range x highest)) <| List.range 0 highest

The `Domino` here is just a type defined by two integers.

But I've messed it up in the inner function so the compiler is yelling

...

a -> Domino

But the left argument is:

Domino

٠.

U2SR9DL7Q: I just added a new comment in beginner channel that adds more clarity. But you're speaking in haskellian right now, and I haven't done that in _awhile_. It's honestly going to take me a few minutes to parse that statement and then I can tell you.

U3SJEDR96 : I think you wnt to move that `)`: `List.map (\x -> List.map (\y -> Domino x y) <| List.range x highest) <| List.range 0 highest`

U48AEBJQ3: https://ellie-app.com/3KQ5Rq7VdHNa1/0?

U2SR9DL7Q: That... worked? It says I've created a list of a list of dominoes. But If I can just flatten that, I should be fine.

U3SJEDR96: `concatMap` to the rescue

U2SR9DL7Q : <@U48AEBJQ3> your solution is probably the more clever, FP way to do it, but I'll have to sit and study it.

U3SJEDR96: or what <@U48AEBJQ3> did, which is nice:slightly_smiling_face:

U3SJEDR96: the observation that for every element in the range, you only want to make combinations of the element and everything that follows is clever: slightly_smiling_face:

U2SR9DL7Q: Yes, but it's very imperative thinking. I've just made the elmy equivalent of ""

for i in range(0, num):

for j in range(i, num):