U4872964V : <@U62UFEG4D> you don't have to make intermediate tuples though, you should be able to apply your function directly
```
points =
    List.map2 Geometry.Point
        (arrayX |&gt; Array.toList)
        (arrayY |&gt; Array.toList)
```

U62UFEG4D : oh wow!
U62UFEG4D : makes totally sense, thank a lot <@U4872964V>!
U4872964V : but now you got to learn uncurry so it's all good :slightly_smiling_face:
U62UFEG4D : exactly hehe!
U0FP80EKB : Don't need to pipeline (personal style, though)
```
points =
    List.map2 Geometry.Point
        (Array.toList arrayX)
        (Array.toList arrayY)
```

U4872964V : yes, that's personal style, I sort of like the "actual" argument being first so that it's visible
U0FP80EKB : Definitely
U5AEH3L05 : Okay, this seems like this should be straightforward: `event.target.value` gives me a float value encoded as a string, so `value : "100"`. How do I decode that into a float?
U5AEH3L05 : If I do that as `Json.Decode.float`, it chokes because of the quotation marks
U5AEH3L05 : My current best approach is
```
    on "input" &lt;|        Json.map tagger &lt;|
            Json.andThen
                (\value -&gt;
                    case String.toFloat value of
                        Ok float -&gt;
                            Json.succeed float

                        Err err -&gt;
                            Json.fail err
                )
                targetValue
```

U236M9FH9 : Decode as a string &amp; use `String.toFloat` with `andThen`, with `fail` as the error case &amp; `succeed` with the success case
U5AEH3L05 : Which seems wildly verbose
U5AEH3L05 : Haha, good timing lysergia :slightly_smiling_face:
U236M9FH9 : You can use `fromResult` from `json-extra`: <http://package.elm-lang.org/packages/elm-community/json-extra/2.3.0/Json-Decode-Extra#fromResult>
U5AEH3L05 : Oh neat, that's certainly terser
U5AEH3L05 : It also looks like parseFloat would work as well
U236M9FH9 : Ah yep, just scrolled down a little :stuck_out_tongue:
U5AEH3L05 : works a charm! thanks <@U236M9FH9>
U30J8RTKJ : Lets say I have a function that accepts 3 arguments, is there a way to partially apply only second argument?
U153UK3FA : <@U30J8RTKJ> the most readable version is to just wrap it in an anonymous function
U153UK3FA : eg `\x y -&gt; yourFunction x 5 y`
U153UK3FA : by convention function args are ordered so that this is a rare thing to need to do
U0CLDU8UB : In the case of the second argument, you could also use flip: `(flip yourFunction) 5` switches the first and the second argument around and then applies `5`.
U0CLDU8UB : But I do agree that the anonymous function is much more readable.
U4872964V : Another option still is to make a helper function with the arguments reversed. If you name stuff well, then this is a very good option.
U4872964V : If it's your own function, maybe changing the order of arguments in the original funciton is a way to go too.
U2DF486HL : Hello, I need to submit a form, but its action is an external POST url.do i need my own Msg and send the Http.request or is there simple way to just call form.submit() ?

examples?

U4872964V : <@U2DF486HL> do you want to be redirected to the POST response page or do you want to still be in your Elm app?
U2DF486HL : <@U4872964V> exactly, it redirects to the new POST response page..
U4872964V : I usually keep track of the fields in my model, and have no `form` element at all, but if you have a `form` element you need to stop propagation of the "submit" event by doing something like```
[ onWithOptions
    "submit"
    { stopPropagation = True, preventDefault = True }
    (Json.Decode.succeed YourSubmitMessage)
]
```

Then the form will instead send `YourSubmitMessage` to your update function, from which you can then issue the post request.

U4872964V : something like this: <https://stackoverflow.com/questions/36387409/how-to-submit-a-form-in-elm>
U4872964V : (the last answer there)
U4872964V : If you don't have a `form` element, you just bind your message to the `onClick` event on the submit button
U2DF486HL : I saw this link before but I thought that it would not help to solve my issue.. because i've to send the form data to the external url, and just that, redirect to its page

U4872964V : oh, you _want_ to redirect?
U2DF486HL : send the form data to the external action url, and move to there site..
U4872964V : then you can just use a normal html form, no need for anything else
U2DF486HL : i usually did (without elm) with just a submit button and the action attrb with its link
U2DF486HL : <@U4872964V> and in the OnSubmit? Right now I have some thing like this:            Html.form
                [ id "theForm"
                , action "<http://external.url/submit>"
                , method "POST"
                , onSubmit ConfirmPayment
                ]
but ConfirmPayment is Msg with no modifications in my Model
```