

U0CLDU8UB : <@U2D5SAEMN> Looks nice! If you wanted, you could make do without the initialCmd by using programWithFlags instead.

U2D5SAEMN : Thanks! Do you mean calculating today in JS instead of in Elm?

U0CLDU8UB : Yeah. You can pass the date in as a Unix timestamp in the flags, for example.

U2D5SAEMN : Ah I see. But what would the benefit of that approach be? Only dropping the `initialCmd`?

U0CLDU8UB : Yeah I'm not saying that it's a better way to do it, just pointing it out. So if some time later you happen to get in a situation where the data _has to_ be resolved from the start, flags is a way to do that.

U2D5SAEMN : Got it, thanks!

U2D5SAEMN : Do you have any further feedback? For example regarding the data model or making things more concise?

U0CLDU8UB : Nothing major, to be honest. The filters are sort of screaming they want to be stored in a Set, but since union types are not `comparable` you would need to use an extra package so maybe it's not worth the hassle.

U2D5SAEMN : Thanks <@U0CLDU8UB>! I'm glad to hear I'm on the right track :slightly_smiling_face: Why would you use a Set for the filters?

U0CLDU8UB : Set is the data structure to reach for when:- you should only have 0 or 1 of a given value, never two or more

- the order is not important

U0CLDU8UB : Assuming you could store the value you wanted in the set, this is how I would do the branch in update:``

```

    SetTypeFilter filter isActive -&gt;
    let
        newFilters =
            if isActive then
                Set.insert filter model.typeFilters
            else
                Set.remove filter model.typeFilters
    in
    ( { model | typeFilters = newFilters }, Cmd.none )
...

```

U2D5SAEMN : ah you'd store the active filters in a set?

U0CLDU8UB : Yep. And the checkbox tells which way it's been turned, so I'd use that information:``

```

type Msg
    = SetTypeFilter EventTypeFilter Bool
-- ...

-- checkbox
input [ type_ "checkbox", checked isChecked, onCheck (SetTypeFilter BirthdayFilter) ] []
...

```

U2D5SAEMN : Nice. But this would require some extra package because union types can't be stored in a Set, right?

U0CLDU8UB : Yes, that's right.

U0CLDU8UB : Though the "magic" of it is that you can use `toString MyType` to get `"MyType"`, which is comparable.

U0CLDU8UB : So if you're interested in learning some data structure stuff, you could try making your own `EverySet`, which uses the core `Set` underneath, but does the string conversion in between.

U2D5SAEMN : Sounds interesting, I'll give it a try!

U2D5SAEMN : Got to go now, thanks for your valuable insights, highly appreciated!

U5QJW0DDE : so, i know Elm cannot directly access the DOM. But, can you give an Elm "element" an id that the DOM can independently find with some raw JS also on the page?

U0LPMPL2U : You can set any attributes you want on DOM elements in Elm

U0LPMPL2U : for example `div [id "my-id", class "my-class"] []`