

U3SJEDR96 : it's `&&`, <@U3YDBDP4Z>

<<http://package.elm-lang.org/packages/elm-lang/core/5.1.1/Basics#&&>>

U3YDBDP4Z : "Cannot find variable !=" :confused:

U3SJEDR96 : `/=`` in elm

U3SJEDR96 : though if you're still dealing with `Maybe`, it's often more readable to use `Maybe.map` and - possibly - `Maybe.withDefault` to fallback to a default value.

U3YDBDP4Z : If I have these types,``

```
type PrivateRoute
```

```
  = HomeRoute
```

```
type PublicRoute
```

```
  = AboutRoute
```

```
  | NotFoundRoute
```

```
  | SignInRoute
```

```
type Route
```

```
  = PrivateRoute
```

```
  | PublicRoute
```

```
``
```

can't `PrivateRoute` and `PublicRoute` be matched as `Route`? :confused:

Trying to put together something that allows me to use a single `case route of` to work with private or public routes

U3YDBDP4Z : In case of a private route, I need to do something. In case of public route, I need to do something else

U3SJEDR96 : You can't "sum" types like that, but you can tag them - `type Route = Private PrivateRoute | Public PublicRoute`

U3SJEDR96 : you can use these in a single case, btw:``

```
case route of
```

```
  Private HomeRoute -&gt; ...
```

```
  Public AboutRoute -&gt; ...
```

```
  Public NotFoundRoute -&gt; ...
```

```
  Public _ -&gt; ...
```

```
``
```

U3YDBDP4Z : Thanks again

U1AN4JRFV : jxxcarlson: Thank you for spelling that out for me. I did notice that it didn't update the correct document, but I thought that could not have anything to do with that, but was intended. And then I had to suddenly run. Did you get help?

U3YDBDP4Z : If `Models` has this type,``

```
type Route
```

```
  = RouteName PrivateRoute
```

```
``
```

Why does an import of `Models` and the use of `Models.Route` somewhere yield a "Models does not expose Route" error?

U0JL9RPC4 : You can expose an union type outside of your module but by default, it does not expose its constructors

U0JL9RPC4 : you have to make something like `export (Route(..)`

U3YDBDP4Z : I had that but with the same problem

U5LHS71SM : Hey, I have a question:I need to display a svg generated by a .js file, I guess I should use ports but I couldn't actually think how to use ports with that? Thanks for the help in advance!

U14Q8S4EM : I can think of two solutions <@U5LHS71SM>

U14Q8S4EM : One would be the manually append it to an html element generated in Elm.

U14Q8S4EM : So in elm make `div [id "svg-container"] []` and in JS write `var svgContainer = document.getElementById("svg-container")` and append your svg to `svgContainer`

U14Q8S4EM : The other would be to convert the svg to json, and then pass it through the port, and the decode it in Elm

U14Q8S4EM : `<svg><circle fill="blue"></circle></svg>` -> `{tag:"svg", children: [{tag:"circle",`

fill:"blue", children: [] }] }

U14Q8S4EM : The first would be far easier, but it might be unstable, and kind of hacky. Because the Elm virtual dom might whip it away if it has reason to update the svg's parent element.

U4F64AKQV : <@U5LHS71SM> Could you generate the SVG in Elm instead using [elm-lang/svg](<<http://package.elm-lang.org/packages/elm-lang/svg/latest>>)?

U0JFGGZS6 : I take it the svg is dynamically generated? Otherwise you could save it to a file and `use xlink:href=...`

U0EUHKVGB : You can also use this script to convert it to elm-svg if it's a statically generated SVG file:

<<https://gist.github.com/eeue56/d0ab060828bf3b64922ac9172ee430c3>>

U5LHS71SM : Thank you all for the answers, I've solved the problem :smile:

U1L4GLFJ6 : hello all

U1L4GLFJ6 : ``module `Html.Attributes` does not expose `Attribute`
import Html.Attributes exposing (style, Attribute)``

U1L4GLFJ6 : how does the `Attributes` library not expose `Attribute`

U0JL9RPC4 : <@U1L4GLFJ6> because `Attribute` is what we call an "opaque" type

U0JL9RPC4 : <<https://stackoverflow.com/questions/40794368/what-is-an-opaque-type-in-elm-and-why-is-it-valuable>>

U0JL9RPC4 : It means you shouldn't manipulate an `Attribute` _directly_ but through some exposed functions

U57KYFW67 : Opaque types are essentially what C++ and Java would call abstract types

U57KYFW67 : You never call constructors for them directly and you never pattern-match on them with `case` directly. You have to go through the public API to work with them.

U57KYFW67 : It's like "all rights reserved" for the implementation, since the library owner can make changes without fear of breaking any existing code

U1L4GLFJ6 : OK. `Svg` exposes itself. so does `Html` but `Attributes` doesn't not. Fine

U57KYFW67 : I can't vouch for the design choice on `Html`, but it's a bit odd to see that it does.