U4872964V : <@U41CTMR1S>, elm-css is closer to plain css, elm-style is a more composable way to do css

U4872964V : <@U2M39B7TM> you'd probably use the `oneOf` decoder, and then a list of `field` decoders with the different constructors

U0LPMPL2U : Look into `Json.Decode.keyValuePairs`

U3HQVHERX : You might want to use `at ["Home","title"]` to dive into the json

U4872964V : `keyValuePairs` does not really work if the values are different types though

U3HQVHERX : <@U41CTMR1S> if you are just starting out i would recommend sticking with plain css

U4872964V : I'd do it like this```

```
oneOf
  [ map Home (field "Home" homeDecoder)
  , map Dresses (field "Dresses" dressesDecoder)
  ]
```

U41CTMR1S : <@U4872964V> <@U3HQVHERX> thanks

U41CTMR1S : <@U3HQVHERX> how can I integrate css file with elm-reactor?

U2M39B7TM : I'll try. ty

U4872964V : <@U41CTMR1S> you can include a stylesheet in your view like this:```

```
node "link"
      [ rel "stylesheet"
      , href "yourstyle.css"
      ]
      []
```

U41CTMR1S : I see. My specific use case is 3rd party widget. Inline css help me to avoid class/id conflict, but feel more complicated and less natural with elm-css/style

U4872964V : not sure what you mean by 3rd party widget

U41CTMR1S : I write widget that another website implement in their website. Disqus is one example.

U4872964V : ok, and they want to change styles with stylesheets?

U41CTMR1S : Usually widgets like Disqus use iframe, but I must seat in the website html for my use case

U41CTMR1S : <@U4872964V> nope. I afraid to collide with their styles

U4872964V : ok, i'd either use a unique class/id prefix or use inline styles

U62JFJWKT : module Calendar exposing (..) &lt;- it doesn't export record types ?

U4872964V : <@U62JFJWKT> it exposes everything, so yes it does

U62JFJWKT : :open_mouth: I suppose i've done a typ

U3HQVHERX : <@U41CTMR1S> you could also use inline styles from the html attributes package. nothing wrong with that. `p [style [("background","red")]][text "hello"]` or instead of using elm reactor you could use `elm-make` and include your js and css on a seperate index page

U3HQVHERX : if you are writing a widget you will likely be embedding it anyway, right?

U41CTMR1S : currently I use React+Styled Components, so I got single js file with inline css, much simpler to implement..

U63UF49NH : In cases like that I really wish Firefox &amp; Edge would finish their ShadowDom implementations.

U63UF49NH : This is better than it has been, but not quire there yet: <http://caniuse.com/#search=shadow%20dom%20v1>

U63UF49NH : I don't know what Elm integration would look like, but I imagine there would be an elegant way to make that work.

U41CTMR1S : <@U3HQVHERX> with html package I loose the prefixing. Also, the official docs say "the general recommendation is to use this function lightly"

U4872964V : <@U41CTMR1S> you can make inline styles with elm-css too

U4872964V : `Css.asPairs`

U41CTMR1S : yep, but I loose pseudo-classes support

U41CTMR1S : looks like no all-in-one solution like styled-compoents but I can live with it

U4872964V : well, then you'd use elm-css to generate the css file (with your prefix of choice)

U41CTMR1S : I'll try it out

U41CTMR1S : thanks again

U4872964V : yeah, how you handle css will depend very much on your use case

U4872964V : that's why there are multiple options, i suppose

U2W6RTGQH : we can import attributes from `Html.Attributes` but exposing `Attributes` from `Html` says `Html doesn't expose Attributes`, then how does `Html.Attributes` work ?

U0LPMPL2U : The _module_ is named `Html.Attributes` but the _type_ is `Html.Attribute`
U0LPMPL2U : note the singular vs plural
U0LPMPL2U : so you could say:```
import Html.Attributes exposing(Attribute)
```

U4872964V : <@U2W6RTGQH> `Html.Attributes` is just a module name. It's nothing that lives in the `Html` module. They are completely unrelated (except that one may import the other).
U2W6RTGQH : <@U0LPMPL2U> <@U4872964V> thanks
U218QNMR8 : Does anyone know why i'd be able to `curl <http://localhost:8000>` when using `elm-reactor` but when opening in the browser it just refuses to connect/times out?
U1C5N3NUD : Hey everybody, is there an easy way of changing pixel values in a `&lt;canvas&gt;`?
U218QNMR8 : have you looked at <http://package.elm-lang.org/packages/evancz/elm-graphics/1.0.1/Collage> <@U1C5N3NUD> ?
U0LPMPL2U : As with HTML, you don't mutate the canvas but instead re-render based on a different model state
U1C5N3NUD : Ah yep it looks like Collage is what I'm looking for
U1C5N3NUD : <@U0LPMPL2U> yeah I just need to change individual pixels on state change
U0LPMPL2U : You'd define a view function that takes in a model and renders the desired pixels
```