U64M3DF1B : funny you mention it - I've totally forgot about it
U64M3DF1B : the console.log says nothing !?!
U41NK9BM4 : Really weird.
U3SJEDR96 : can you explain _what_ goes wrong? Also, where do you use `ports.js` and how? Does `elm-make Main.elm --output elm.js` give any output at all?
U64M3DF1B : if I change Main.elm to```
import Html

main = Html.text "hello"
```

U64M3DF1B : upps
U3SJEDR96 : yeah, that should do the trick. Well, that and having `module Main exposing (..)` at the very top
U3SJEDR96 : as in, that's a valid module
U64M3DF1B : all is well
U3SJEDR96 : On a general note, I would focus on learning how to use Elm before focusing too much on having it run both in electron _and_ the browser. Getting it running in the browser first, should make it quite easy to getting it to run in electron after
U64M3DF1B : ok - so now I am really confused!
I tried your suggestion - that works!

```
module Main exposing (..)
import Html
main = Html.text "hello world"
```

U3SJEDR96 : (lacking a module at the top will default to `module Main`, afaik)
U64M3DF1B : but
if I do this:

```
module Main exposing
  ( Model
  , Msg
  , init
  , update
  , subscriptions
  , view
  )


```

stops the fun! And leaves me with `..embed bla bla undefined`

U3SJEDR96 : But do you have a `Model`, `Msg`, and those functions defined?
U64M3DF1B : so - under the cover "exposing (..)" kind a _spills_ the *embed* too?
U3SJEDR96 : `exposing` only declares what functions any other Elm module can import from your module
U3SJEDR96 : `embed` is enabled by having a function called `main` with a valid signature
U64M3DF1B : that - I would chalk up as _a feature_
U3SJEDR96 : note that `main` doesn't _have to be_ exposed, but it can.
U3SJEDR96 : but you can only `expose` things that are actually defined in a given module
U3SJEDR96 : so if you haven't written those functions and types, you can't compile it
U64M3DF1B : that I understand