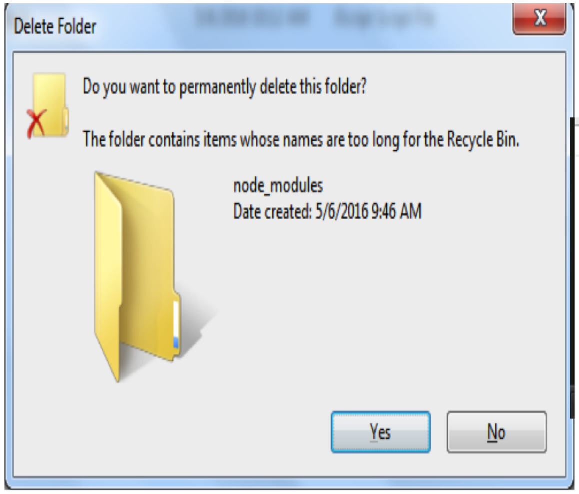
A MOŻE BY TAK RZUCIĆ TO WSZYSTKO I NAUCZYĆ SIĘ ELMA

https://przemyslawjanpietrzak.github.io/przemyslawjanpietrzak.github.io/elm/dist

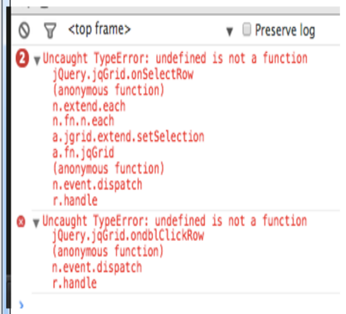


Delete Folder

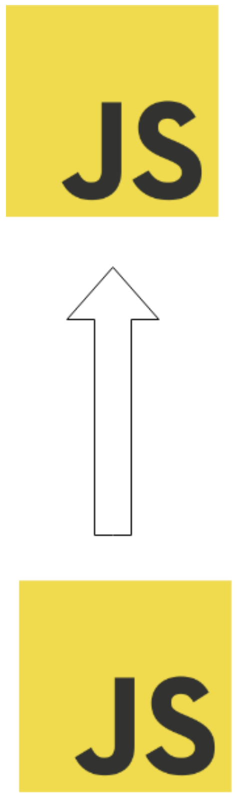
Do you want to permanently delete this folder?

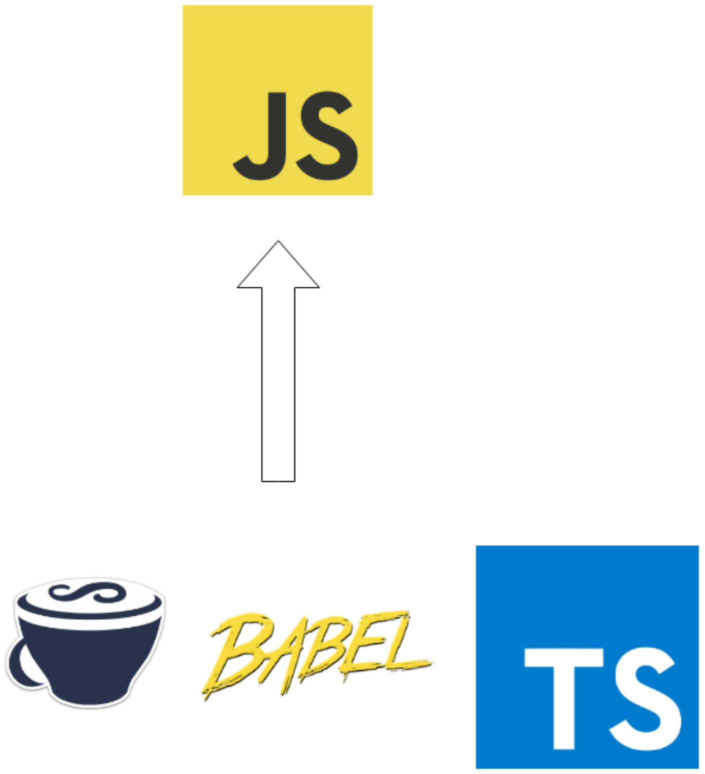
The folder contains items whose names are too long for the Recycle Bin.

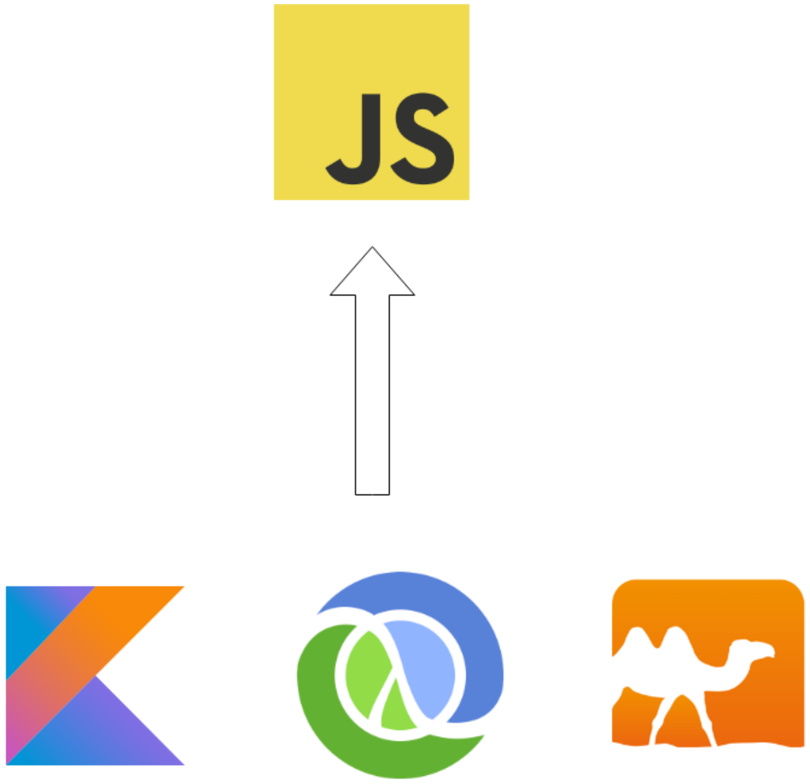
node_modules
Date created: 5/6/2016 9:46 AM

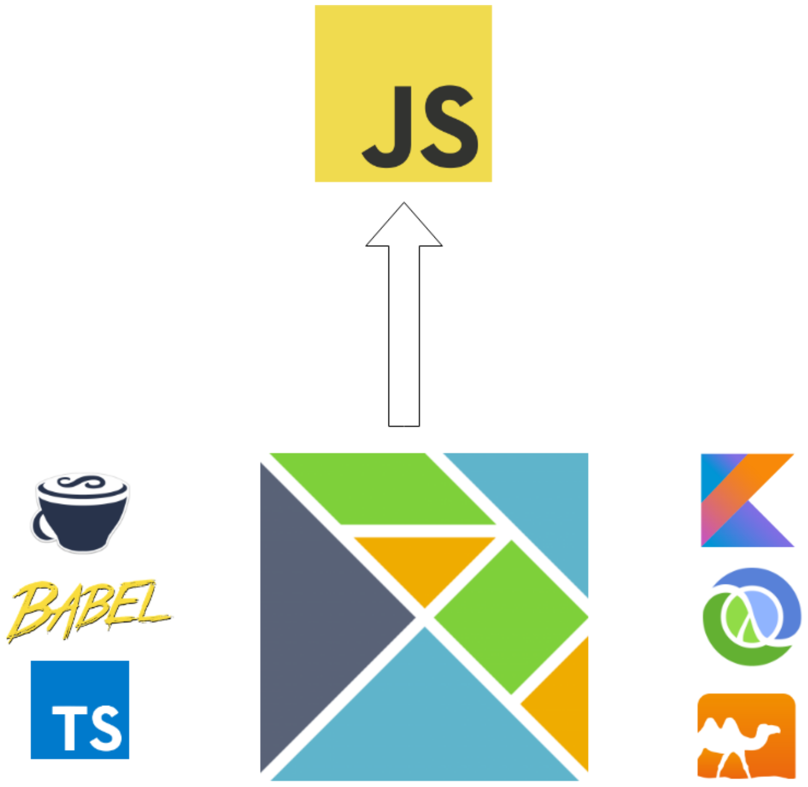


```
Uncaught TypeError: undefined is not a function
  jQuery.jqGrid.onSelectRow
  (anonymous function)
  n.extend.each
  n.fn.n.each
  a.jgrid.extend.setSelection
  a.fn.jqGrid
  (anonymous function)
  n.event.dispatch
  r.handle
Uncaught TypeError: undefined is not a function
  jQuery.jqGrid.ondblClickRow
  (anonymous function)
  n.event.dispatch
  r.handle
```

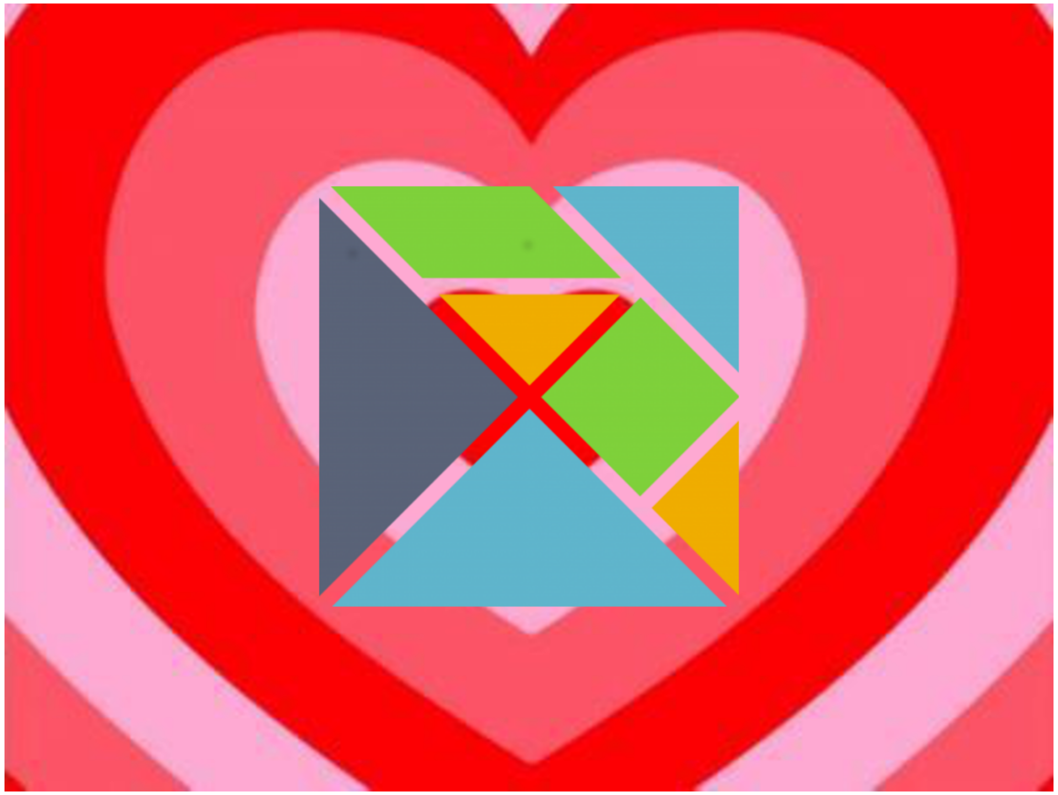

JS
JS


JS
BABEL
TS


JS


JS
BABEL
TS

SUGAR



ELM TIME

```javascript
const add = (x, y) => x + y;

[1,2,3,4]
[1].concat([2,3,4])
[1].concat([2]).concat([3]).concat([4])

if (maybeList.value) {
  return maybeList.value;
} else {
  return [];
}

const point = { x: 3, y: 4 }
const origin = { x: 0, y: 0 };
[ origin, point ].map(({ x }) => x)
{ ...point, x: point.x + 1, y: point.y + 1 }

const squares = range(0, 1000).map(i => i ** 2)
```

```elm
add x y = x + y

[1,2,3,4]
1 :: [2,3,4]
1 :: 2 :: 3 :: 4 :: []

case maybeList of
  Just xs -> xs
  Nothing -> []

point = { x = 3, y = 4 }
origin = { x = 0, y = 0 }
List.map .x [ origin, point ]
{ point | x = point.x + 1, y = point.y + 1 }

squares =
  List.map (\n -> n^2) (List.range 1 100)
```

LET IT BE

```
const square = x => x ** 2;
square(2); // 4

const doubleAndIncrease = (x) => {
  const y = x * 2;
  return y + 1;
};
doubleAndIncrease(2) // 5
```

```
square x = x ^ 2
square 2 -- 4

doubleAndIncrease x =
  let
    y = x * 2
  in y + 1
doubleAndIncrease 2 -- 5
```

PIPE FICTION

```
const calculate = (x) => {
  return square(
    add(
      1, multiply(2, x)
    )
  );
};
```

```
let calculate x =
    |> multiply 2
    |> add 1
    |> square
```

```
wc(grep('waring', lint()))
```

```
#/bin/bash

npm run lint | grep warring | wc -l
```

HASKELL CURRY (1900 - 1982)

```
const sumThree = x => y => z => x + y + z;


const sumTwo = sumThree(0);


const addFour = sumTwo(4);


addFour(5) // 9


sumTwo(4)(5) // 9
```

```
sumThree : Int -> Int -> Int -> Int
sumThree x y z = x + y + z
-- <function> : number -> number -> number -> number

sumTwo = sumThree 0
-- <function> : number -> number -> number

addFour = sumTwo 4
-- <function> : number -> number

addFour 5
-- 9 : number

sumTwo 4 5
-- 9 : number
```

PATTERN MATCHING

```
case maybeList of
  Just xs -> xs
  Nothing -> []

case xs of
  [] ->
    Nothing
  first :: rest ->
    Just (first, rest)

case n of
  0 -> 1
  1 -> 1
  _ -> fib (n-1) + fib (n-2)
```

UNION TYPES

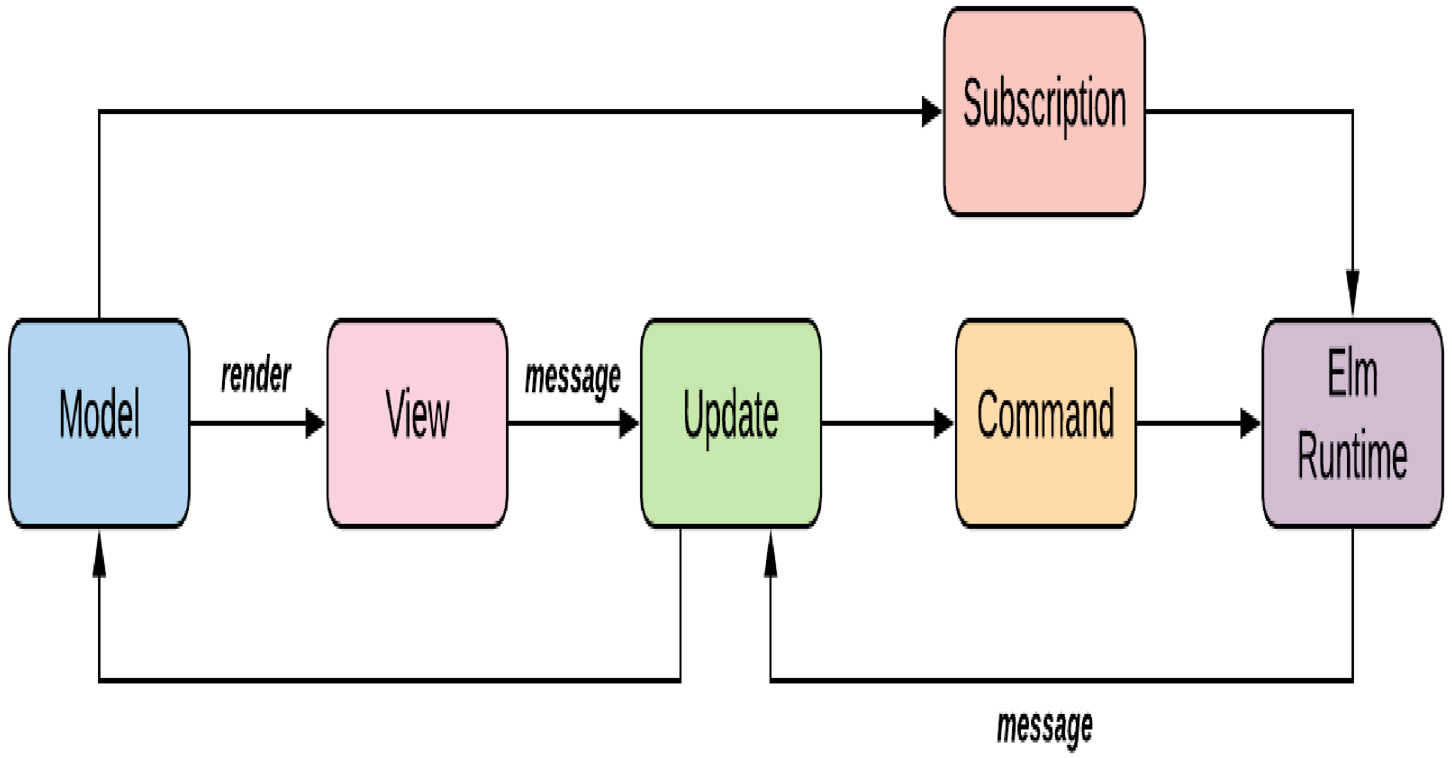
```
type MyThing
AThink  = AString String
  | AnInt Int
  | ATuple (String, Int)

unionFn : MyThing -> String
unionFn thing =
  case thing of
    AString s -> "It was a string: " ++ s
    AnInt i -> "It was an int: " ++ toString i
    ATuple (s, i) -> "It was a string and an int: " ++ s ++ " and " ++ toString i
```

ELM FRAMEWORK

```
const reducer = (state = 0, action) => {
  switch (action.type) {
    case 'Increment':
      return state + 1;
    case 'Decrement':
      return state - 1;
  }
}

class View extends React.component {
  render() {
    return <div>
      <button onClick={this.decrement.bind(this)}> - </button>
      <div>{ this.model }</div>
      <button onClick={this.increment.bind(this)}> + </button>
    </div>
  }
}
```

```
main =
  Browser.sandbox { init = 0, update = update, view = view }

type Msg = Increment | Decrement

update msg model =
  case msg of
    Increment ->
      model + 1

    Decrement ->
      model - 1

view model =
  div []
    [ button [ onClick Decrement ] [ text "-" ]
    , div [] [ text (String.fromInt model) ]
    , button [ onClick Increment ] [ text "+" ]
    ]
```

ELM ARCHITECTURE



BATTERIES INCLUDED

WHY ELM?

- No runtime error
- Testable
- Batteries included
- Organized
- Easy debug
- Light and fast
- Compilator hints

"DOBRY JĘZYK NIE POZWALA CI PISAĆ DOBREGO KODU, TYLKO ZABRANIA PISANIA ZŁEGO"

zasłyszane w Fat Bobie

ELMENTARY DEAR WATSON

- Why ELM?
- Making Impossible States Impossible
- Convergent Evolution

ELM ARCHITECTURE IN JS

link

```
// redux-loop
function reducer(state = initialState, action) {
  switch(action.type) {
  case 'INIT':
    return loop(
      {...state, initStarted: true},
      Cmd.run(fetchUser, {
        successActionCreator: userFetchSuccessfulAction,
        failActionCreator: userFetchFailedAction,
        args: ['123']
      })
    );

  case 'USER_FETCH_SUCCESSFUL':
    return {...state, user: action.user};

  case 'USER_FETCH_FAILED':
    return {...state, error: action.error};

  default:
    return state;
  }
}
```

ELM ARCHITECTURE IN REASON

link

```
 /* rembrandt */
open Rembrandt.Elements;
type model = int;
type action =
  | Add
  | Sub
  | Twice;

let update =
    (model: model, action: action): (model, Command.command('action)) =>
  switch (action) {
  | Add => (model + 1, Command.null)
  | Sub => (model - 1, Command.null)
  | Twice => (model + 1, Command.action(Add))
  };

Rembrandt.run(
  ~model=42,
  ~update,
  ~view=
    (model, dispatch) =>
      <div>
        <div id="count"> {string_of_int(model) |> text} </div>
        <button id="plus" onClick={_ => Add |> dispatch}>
          {text("+")}
        </button>
        <button id="minus" onClick={_ => Sub |> dispatch}>
          {text("-")}
        </button>
        <button id="double" onClick={_ => Twice |> dispatch}>
          {text("twice +")}
        </button>
      </div>,
  (),
);
```

ANY QUESTIONS?

CSS IN ELM

```elm
-- elm-css
logo : Html msg
logo =
    img
        [ src "logo.png"
        , css
            [ display inlineBlock
            , padding (px 20)
            , border3 (px 5) solid (rgb 120 120 120)
            , hover
                [ borderColor theme.primary
                , borderRadius (px 10)
                ]
            ]
        ]
        []
```

```elm
-- elm-stylesheet
main = myStylesheet : Stylesheet
myStylesheet =
  let
    myClassStyles =
      newRuleSet
        |> withSelectors
          [ Class selectors.myClass ]
        |> withDeclarations
          [ ("font-family", FontStack fonts)
          , ("font-weight", Num weight.normal)
          , ("font-size", Unit 2 Em)
          , ("color", Col palette.blue)
          , ("text-align", Str "center")
          ]
```

THANK YOU :*