

Elm na produkcję!

...serio, ma działać jutro...

Cel

— — —

- obniżenie postrzeganego progu wejścia w hipsterską technologię ELM

Cel

- obniżenie postrzeganego progu wejścia w hipsterską technologię ELM

przez

- szybkie omówienie The Elm Architecture (TEA)

Cel

- obniżenie postrzeganego progu wejścia w hipsterską technologię ELM

przez

- szybkie omówienie The Elm Architecture (TEA)
- skupimy się przy tym na komunikacji TEA z JSem

Cel

- obniżenie postrzeganego progu wejścia w hipsterską technologię ELM

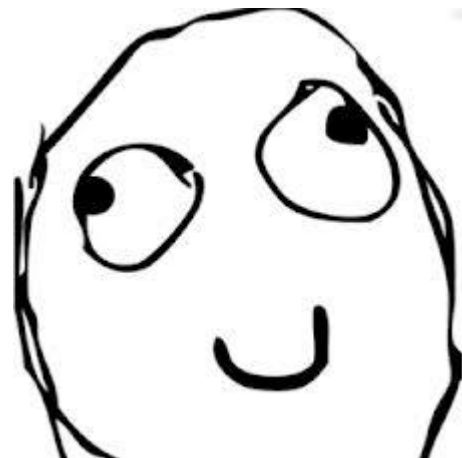
przez

- szybkie omówienie The Elm Architecture (TEA)
- skupimy się przy tym na komunikacji TEA z JSem
- przykłady kodu/konfiguracji elmowego komponentu w 4 różnych stackach JSowych

Co się nie stanie

— — —

- dowiem się tutaj wszystkiego co muszę wiedzieć o ELMie żeby jutro zacząć pisać kod na produkcję



(<https://imgflip.com/s/meme/Derp.jpg>)

Taka sytuacja

— — —

- jesteś developerem w projekcie który robi...

Taka sytuacja

— — —

- jesteś developerem w projekcie który robi...
- appkę TODOMVC

Taka sytuacja

— — —

- jesteś developerem w projekcie który robi...
- appkę TODOMVC
- przychodzi do nas feature request

FOCUS MODE

— — —

1. na liście todosów użytkownik wybiera te na których chce się fokusować

FOCUS MODE

— — —

1. na liście todosów użytkownik wybiera te na których chce się fokusować
2. klika 'focus mode!!'

FOCUS MODE

— — —

1. na liście todosów użytkownik wybiera te na których chce się fokusować
2. klika 'focus mode!!'
3. w focus mode widać tylko taski które wybrał,

FOCUS MODE

1. na liście todosów użytkownik wybiera te na których chce się fokusować
2. klika 'focus mode!!'
3. w focus mode widać tylko taski które wybrał,
4. wracasz do listy dopiero jak 'odhaczysz' wszystkie fokusowane taski

FOCUS MODE

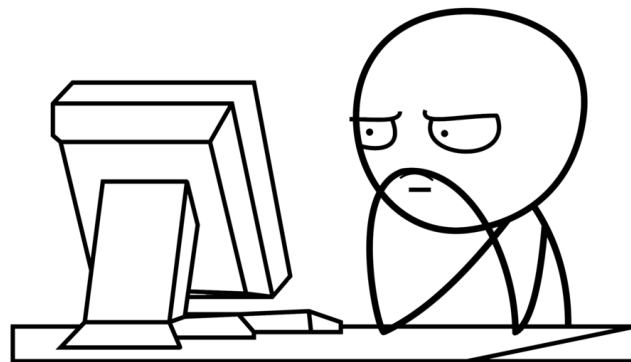
1. na liście todosów użytkownik wybiera te na których chce się fokusować
2. klika 'focus mode!!'
3. w focus mode widać tylko taski które wybrał,
4. wracasz do listy dopiero jak 'odhaczysz' wszystkie fokusowane taski
5. po powrocie taski są odhaczone na naszej todo-liście

BTW...

mamy już gotowy
komponent do focus mode,
wystarczy go użyć...



(<http://i0.kym-cdn.com/photos/images/newsfeed/000/343/462/79a.gif>)

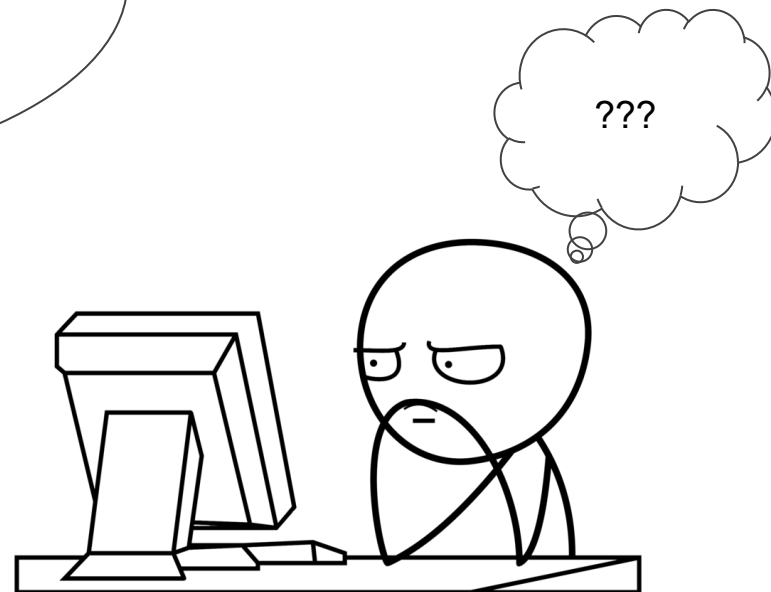


(https://vignette.wikia.nocookie.net/lego/images/f/f2/Hd_computer_guy_meme_by_zapgod16-d4t2jh3.png/revision/latest?cb=20130322044600)

BTW...



(<http://i0.kym-cdn.com/photos/images/newsfeed/000/343/462/79a.gif>)



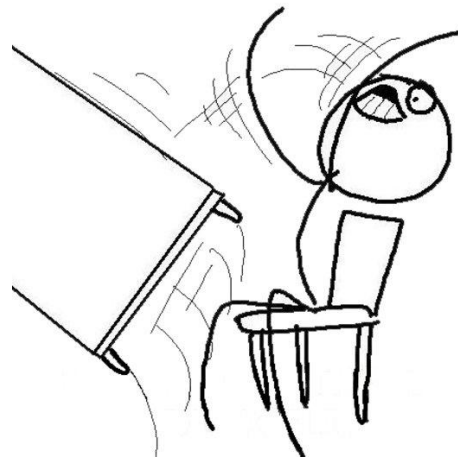
(https://vignette.wikia.nocookie.net/lego/images/f/f2/Hd_computer_guy_meme_by_zapgod16-d4t2jh3.png/revision/latest?cb=20130322044600)

BTW...

...ma
być na
jutro



(<http://i0.kym-cdn.com/photos/images/newsfeed/000/343/462/79a.gif>)



(http://i0.kym-cdn.com/entries/icons/facebook/000/006/725/desk_flip.jpg)

ELM

— — —

- kompiluje się do JSa

ELM

— — —

- kompiluje się do JSa
- jest magiczny, hipsterski, ale ma warstwę komunikacji z kodem JS

ELM

— — —

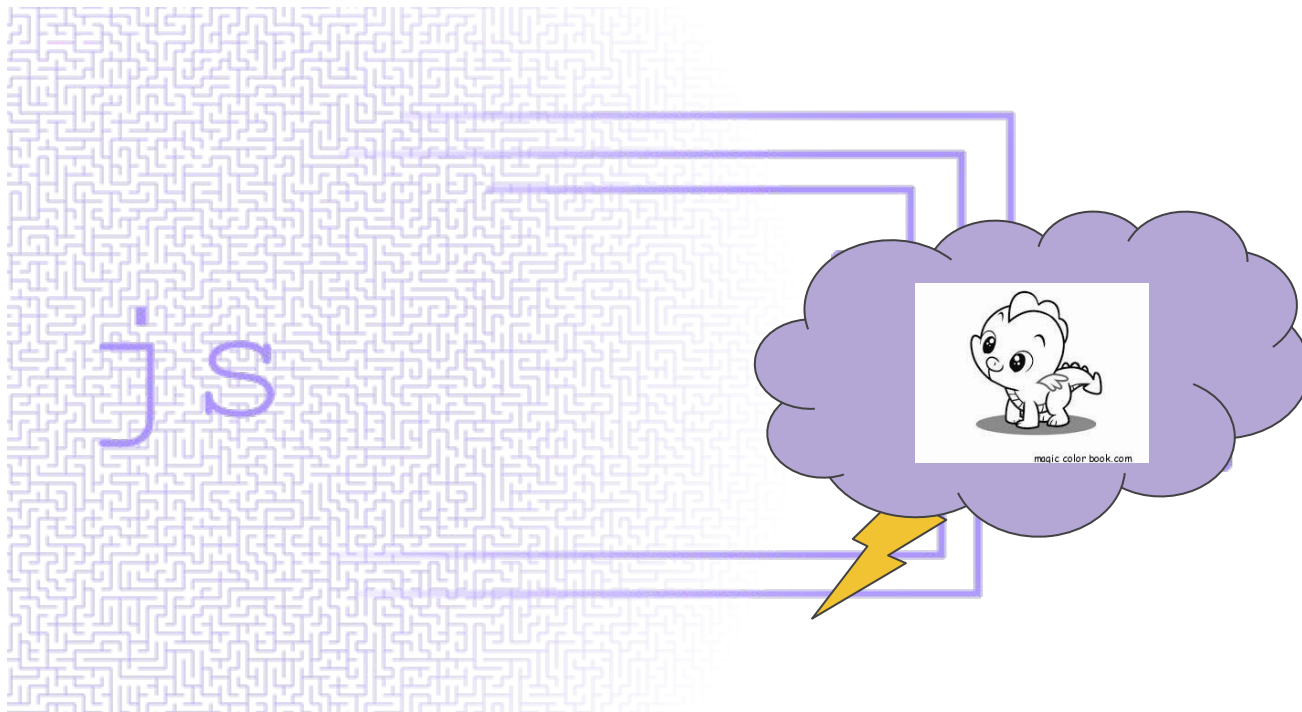
- kompiluje się do JSa
- jest magiczny, hipsterski, ale ma warstwę komunikacji z kodem JS
- PORTY

ELM



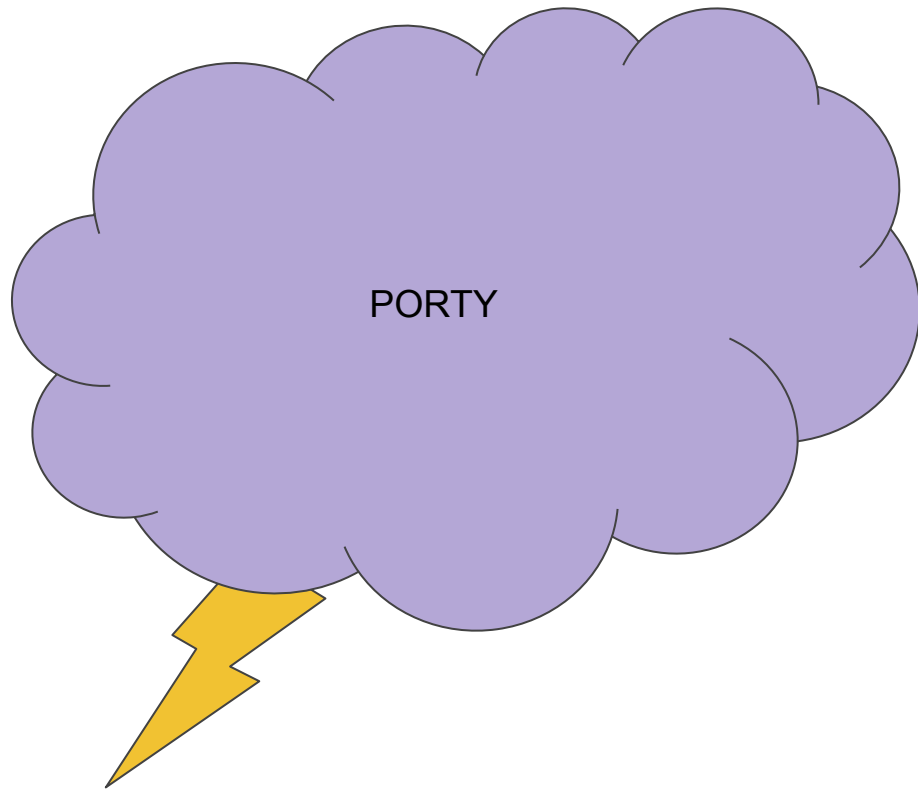
(<https://guide.elm-lang.org/interop/interop.png>)

MAGIC



(<https://guide.elm-lang.org/interop/interop.png>)

Example



chwila moment, komponent?

— — —

chwila moment, komponent?

— — —

- kompozycja jest fajna

chwila moment, komponent?

— — —

- kompozycja jest fajna
- komponenty jako czarne skrzynki są fajne

chwila moment, komponent?

— — —

- kompozycja jest fajna
- komponenty jako czarne skrzynki są fajne
- jasne interfejsy są fajne

chwila moment, komponent?

— — —

- kompozycja jest fajna
- komponenty jako czarne skrzynki są fajne
- jasne interfejsy są fajne
- elm też może być fajny :3

lecimy z kodem

— — —

- vanillajs
- react + redux
- vue
- angular 4(?)

Podsumowując

— — —

Podsumowując

— — —

- aplikacje złożone z komponentów, z definicji, poddają się kompozycji

Podsumowując

— — —

- aplikacje złożone z komponentów, z definicji, poddają się kompozycji
- powinny mieć jasne interfejsy i odpowiedzialności

Podsumowując

— — —

- aplikacje złożone z komponentów, z definicji, poddają się kompozycji
- powinny mieć jasne interfejsy i odpowiedzialności
- w idealnym świecie każdy feature mógłby być w innej technologii

Podsumowując

- aplikacje złożone z komponentów, z definicji, poddają się kompozycji
- powinny mieć jasne interfejsy i odpowiedzialności
- w idealnym świecie każdy feature mógłby być w innej technologii
- technologie jak webpack mogą w tym pomóc

https://github.com/tarnas14/elm_to_prod_tomorrow

<https://guide.elm-lang.org/architecture>

<https://github.com/elm-community/elm-webpack-loader>



github @tarnas14
twitter @tarnas14