Reason ML

Or a journey to learning a new language

VIETNAM
WEB
SUMMIT

Today's talk

- The Joy of Learning
- ReasonML - What and Why
- Code Samples!

VIETNAM WEB SUMMIT

How many here know React?

VIETNAM
WEB
SUMMIT

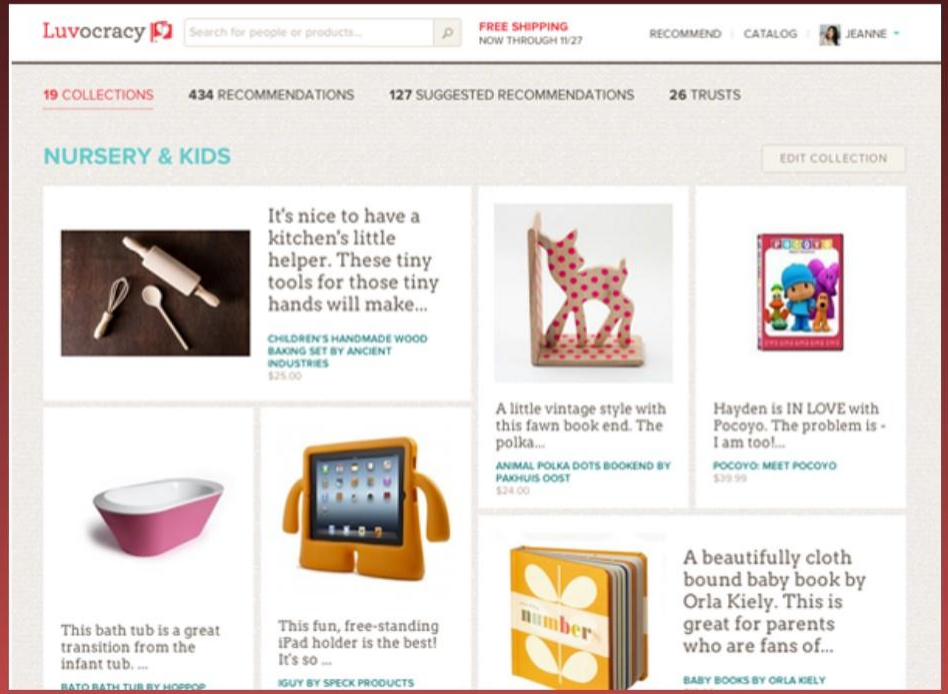
...ok, of those raising your hands,
how many of you know jQuery?

VIETNAM
WEB
SUMMIT

jQuery was my “first” JavaScript Experience

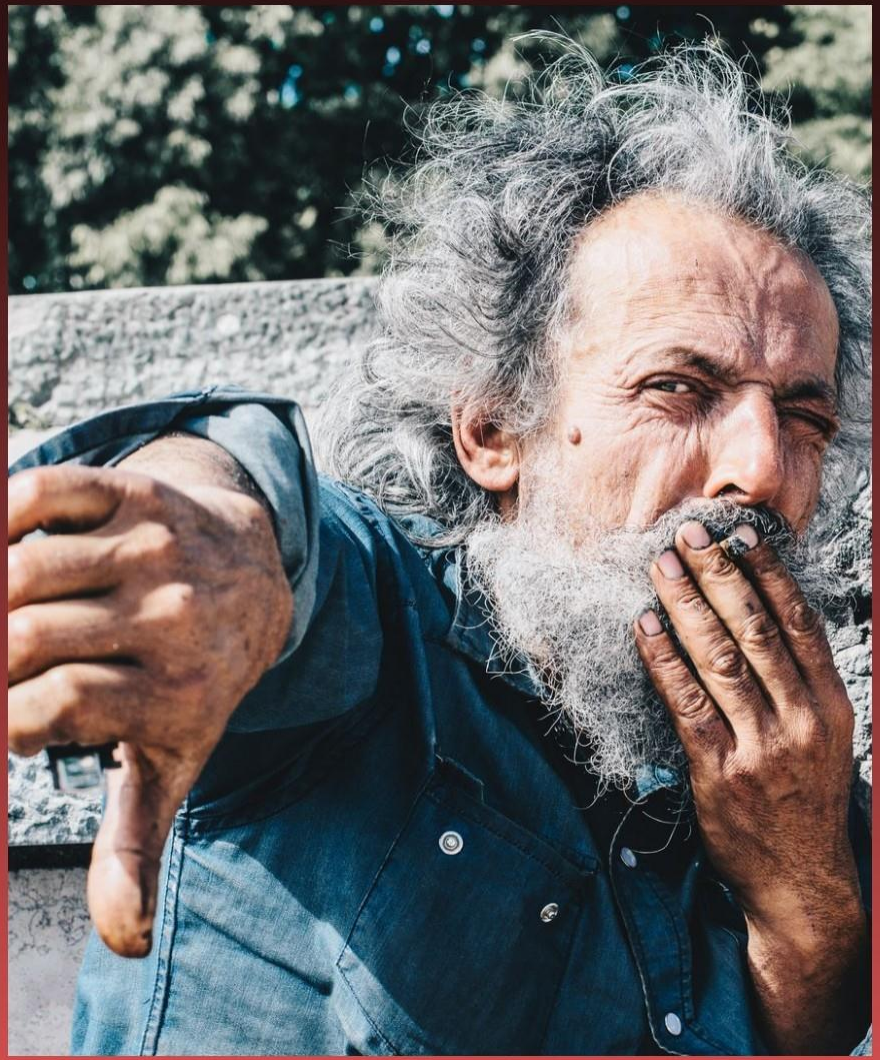
- It was so easy to change small things
- I didn't have to initialize a whole SPA
- Error Messages are pretty simple
- Didn't need any toolchain like Grunt or Babel

Built a Product on jQuery in 2013



- Several million users
- Engineering team of 10
- Pretty interactive
- Built on Rails

VIETNAM WEB SUMMIT



Thinking about jQuery now...

VIETNAM WEB SUMMIT

My next startup, we were a SPA on Backbone.js.

I was hooked!

I don't use it now.

But I'm so glad I learned!

VIETNAM
WEB
SUMMIT

Learn at least one new language every year.

Different languages solve the same problems in different ways. By learning several different approaches, you can help broaden your thinking and avoid getting struck in a rut.

PRAGMATIC PROGRAMMER

VIETNAM WEB SUMMIT

Be a Boss

- Programming languages always change
- It'll help your job prospects if you can stay ahead of the market
- Employers like people who are up to date
- You'll see a lot of the same patterns and learn about different architectures

VIETNAM WEB SUMMIT

Why don't more people do this?

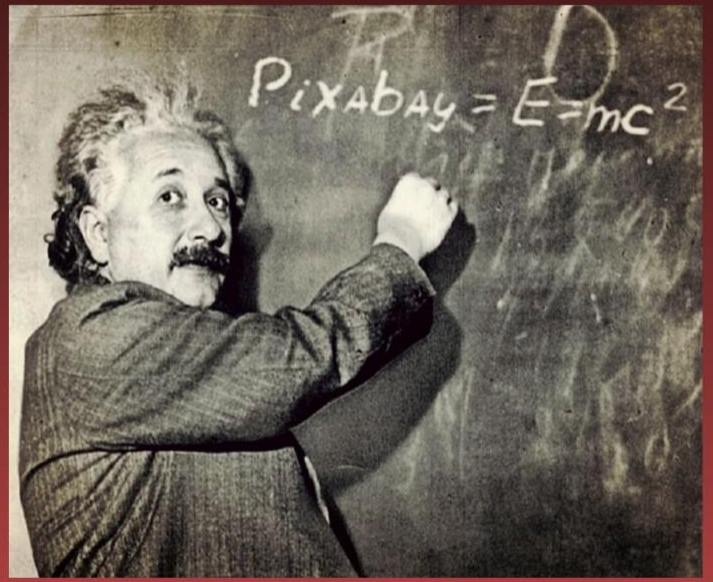
VIETNAM
WEB
SUMMIT

My experience with languages

- toilet o dau
- where is the toilet
- ou sont les toilettes
- donde esta el bano
- wo ist die toilette
- hwajangshil uh deesuhyo
- chimbudzi chili kuti

VIETNAM WEB SUMMIT

Proficiency is important, though!



VIETNAM WEB SUMMIT

10 Steps of Language Learning

(at least mine)

VIETNAM
WEB
SUMMIT

Step 1

I am excited to do this!
What nice documentation!

VIETNAM
WEB
SUMMIT

Step 2

The Get Started Example works perfectly! That was easy.

VIETNAM
WEB
SUMMIT

Step 3

I tried to do something a little different and nothing works now.

VIETNAM WEB SUMMIT

Step 4

Hmm I can't find the exact thing I want in the documentation.

VIETNAM WEB SUMMIT

Step 5

Argh I know exactly how to do this in the old language. Can I go back?

VIETNAM WEB SUMMIT

Step 6

Oh god this is taking me longer than I thought...

VIETNAM
WEB
SUMMIT

Step 7

This will never work. There must be a bug with the framework

Step 8

I am so sick of looking at the same documentation page over and over

VIETNAM
WEB
SUMMIT

Step 9

I just wrote the ugliest code ever trying to do it the old way, but in this language, but it works

VIETNAM WEB SUMMIT

Step 10

I am a god

Languages

OCaml

The real language backing everything. Been around for a long time and very well-respected.

ReasonML

A “flavor” of OCaml. Different syntax.

BuckleScript

BuckleScript compiles OCaml/Reason code into JavaScript.

Why ReasonML?

- **Typing**

  But you'd have that in TypeScript and Flow anyway

- **Functional Programming**

  People tell me it's better... it's more expressive.

Reason Example #1: Sums

```
let rec sum = (numbers) =>
    switch(numbers) {
        | [] => 0
        | [x, ...numbers] => x + sum(numbers)
    };

let () =
    [1, 2, 3, 4, 5]
    |> sum
    |> Js.log

```

VIETNAM WEB SUMMIT

Reason Example #2: FizzBuzz

```
let rec fizzbuzz = (i) =>
    switch((i mod 3, i mod 5) ) {
        | (0, 0) => "FizzBuzz"
        | (0, _) => "Fizz"
        | (_, 0) => "Buzz"
        | _ => string_of_int(i)
    }
let () =
    for (i in 1 to 100) {
      Js.log(fizzbuzz(i));
```

VIETNAM WEB SUMMIT

ReasonReact may be the future of React

- Creator of React originally wrote everything in an ML-language (similar to OCaml)
- Reason is officially maintained by Facebook
- Facebook actually uses it. As of Sept 2017, Messenger was 50% written in Reason!

FB Messenger now 50% written in Reason (last year)

https://reasonml.github.io/blog/2017/09/08/messenger-50-reason

- Full rebuild of the Reason part of the codebase is ~2s
- Messenger used to receive bugs reports on a daily basis; since the introduction of Reason, there have been a total of 10 bugs (that's during the whole year, not per week)!
- **Most new features are now developed in Reason.**
- Refactoring speed went from days to hours to dozens of minutes.

VIETNAM WEB SUMMIT

Reason React Example

Step 1: Defining our Types

```
type photo = {
  albumId: int,
  id: int,
  title: string,
  url: string,
  thumbnailUrl: string
};

type state =
  | Loading
  | Error
  | Loaded(list(photo));

type action =
  | PhotosFetch
  | PhotosFetched(list(photo))
  | PhotosFailedToFetch;

```

VIETNAM WEB SUMMIT

Reason React Example

Step 2: Define how to read our JSON

```
module Decode = {
  let photo = photo =>
    Json.Decode.{
      albumId: field("albumId", int, photo),
      id: field("id", int, photo),
      title: field("title", string, photo),
      url: field("url", string, photo),
      thumbnailUrl: field("thumbnailUrl", string, photo)
    };
  let photos = json : list(photo) => Json.Decode.list(photo,
};son);
```

VIETNAM WEB SUMMIT

Reason React Example

Step 3: Build up our Component

```
1 let component = ReasonReact.reducerComponent("PhotosList");
2
3 let make = _children => {
4   ...component,
5   initialState: _state => Loading,
6
7
```

VIETNAM WEB SUMMIT

Reason React Example

Step 4: Define our Reducer (starting wit the easy ones)

```
1   reducer: (action, _state) =>
2     switch (action) {
3     | PhotosFetched(photos) => ReasonReact.Update(Loaded(photos))
4     | PhotosFailedToFetch => ReasonReact.Update(Error)
```

Reason React Example

Step 4: Call our API

```
| PhotosFetch => ReasonReact.UpdateWithSideEffects(
    Loading,
    (
      self =>
        Js.Promise.(
          Fetch.fetch("https://jsonplaceholder.typicode.com/Photos")
          |> then_(Fetch.Response.json)
          |> then_(json =>
               json
               |> Decode.photos
               |> (photos => self.send(PhotosFetched(photos)))
               |> resolve
             )
          |> catch(_err =>
               Js.Promise.resolve(self.send(PhotosFailedToFetch))
             )
```

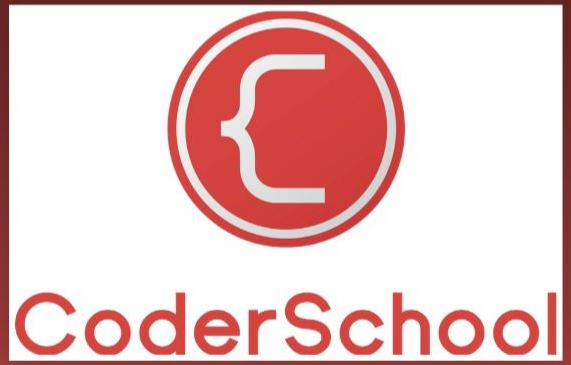
Reason React Example

Step 5: Write our Render function

```
didMount: self => self.send(PhotosFetch),
  render: self =>
    switch(self.state) {
      | Error => <div> (s("Something went wrong")) </div>
      | Loading => <div> (s("NOW LOADING")) </div>
      | Loaded(photos) =>
        <div>
          <h1> (s("Photos")) </h1>
          <ul>
            (
              List.map(photo =>
                <li key=string_of_int(photo.id)>
                  <p> (s(photo.title)) </p>
                  <img src=photo.thumbnailUrl/>
                </li>
              , photos)
              |> Array.of_list
```

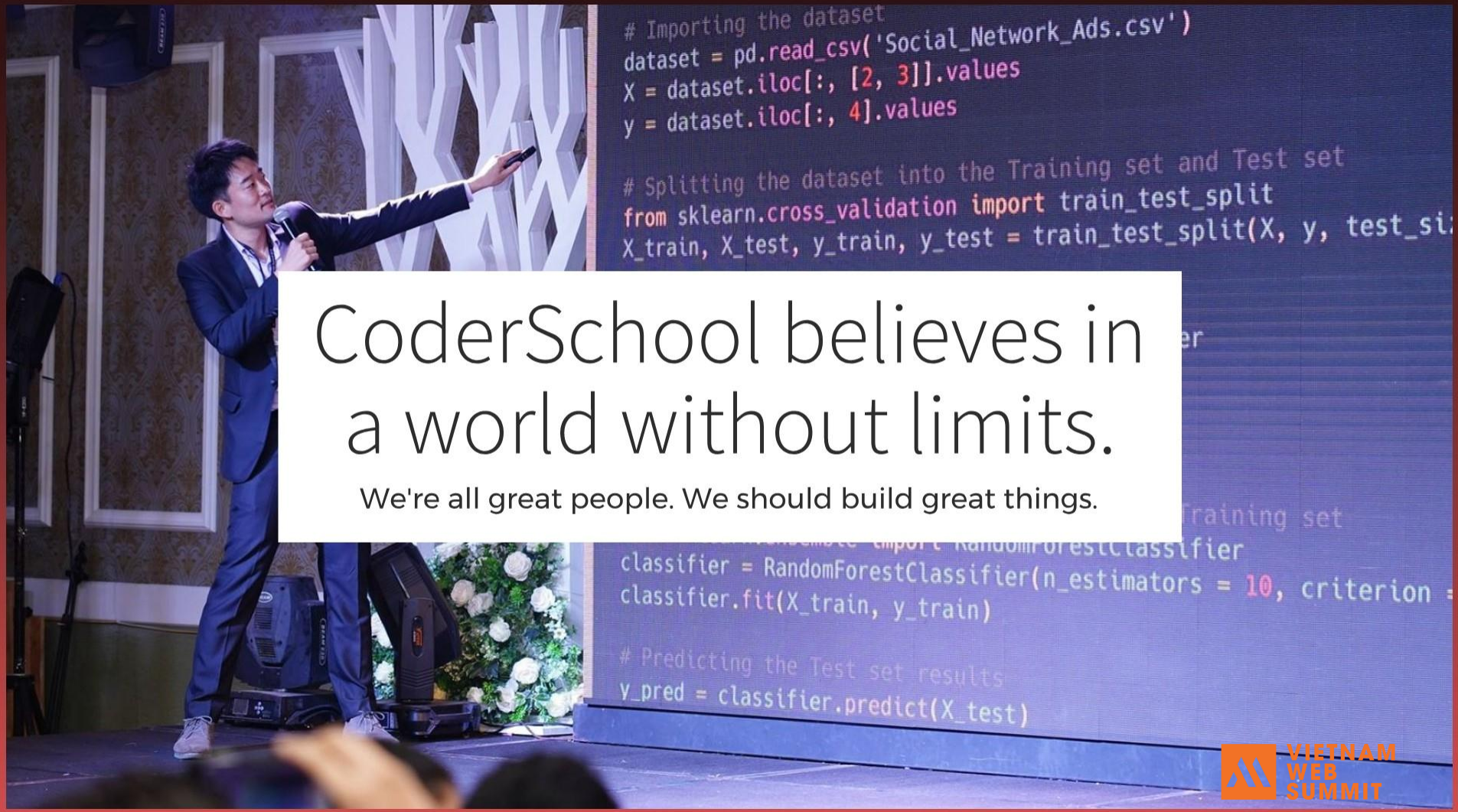
CoderSchool

12 Ton Dan, District 4 - www.coderschool.vn



- Founded in 2015, with over 500 graduates.
- Curriculum developed in Silicon Valley, Vietnam, and Singapore
  - Machine Learning
  - React Native
  - UX
- We want developers to be happy.

VIETNAM WEB SUMMIT


CoderSchool believes in a world without limits.
We're all great people. We should build great things.
# Importing the dataset
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_si
classifier = RandomForestClassifier(n_estimators = 10, criterion =
classifier.fit(X_train, y_train)
# Predicting the Test set results
y_pred = classifier.predict(X_test)
VIETNAM WEB SUMMIT