ALEXANDER VARWIJK - AUGUST 17TH/18TH 2022

FAST, SIMPLE, FULLY TYPED JAVASCRIPT FROM THE FUTURE

WHOAM(I)

ALEXANDER VARWIJK

- Lead Front-End Engineer at

- 20 Years of programming
- 10+ years professionally

- Addicted to coffee
- Having a blast in NYC

IMAGINE

TYPESCRIPT

JavaScript with syntax for types

Photo by Jens Lelie on Unsplash

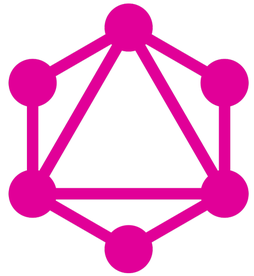

rescript

WHO IS USING RESCRIPT?

WHY DID OPEN SOCIAL CHOOSE RESCRIPT?

~~Mistakes~~

Type Inference ✓



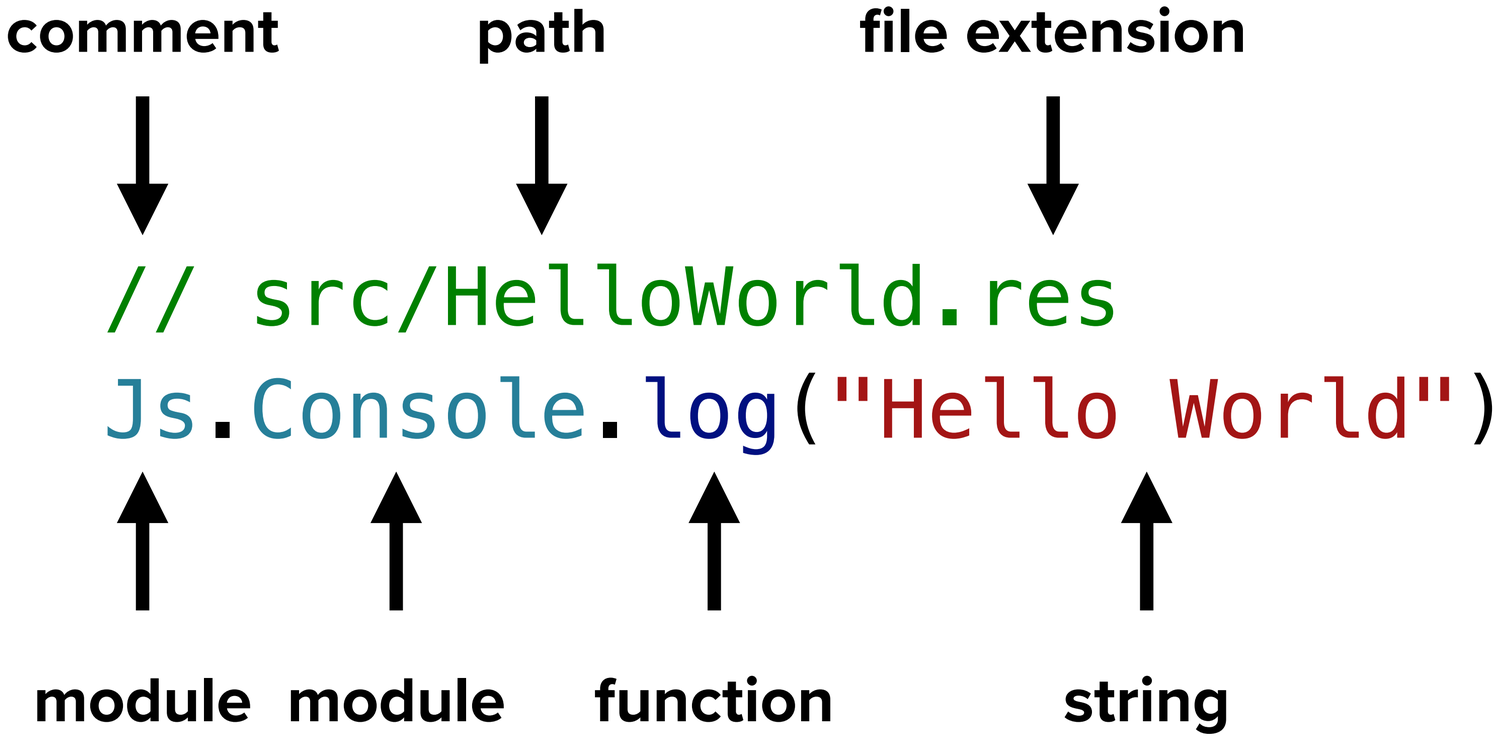
open social®

```
npm install rescript
```

UPDATE PACKAGE.JSON

```
"scripts": {
  "build": "rescript build",
  "watch": "rescript build -w"
}
```

WHAT DOES IT LOOK LIKE?



BUILD

```
$ npm run build

> rescript build

Error: bsconfig.json: No such file or directory
```

ADD BSCONFIG.JSON

package name →

input →

output code →

output file extension →

```
{
  "name": "hello-world",
  "sources": [
    {
      "dir": "src",
      "subdirs": true
    }
  ],
  "package-specs": [
    {
      "module": "es6",
      "in-source": true
    }
  ],
  "suffix": ".bs.js"  // one of: .js, .mjs, .cjs, .bs.js
}
```

BUILD

```
$ npm run build

> rescript build

rescript: [3/3] src/HelloWorld.cmj
```

OUTPUT

```
// src/HelloWorld.bs.js
// Generated by ReScript, PLEASE EDIT WITH CARE


console.log("Hello World");

export {

}
/*  Not a pure module */
```

```
npm install @rescript/react react react-dom
```

UPDATE BSCONFIG.JSON

```
{
  "name": "hello-world",
  "sources": [
    {
      "dir": "src",
      "subdirs": true
    }
  ],
  "package-specs": [
    {
      "module": "es6",
      "in-source": true
    }
  ],
  "suffix": ".bs.js",
  "reason": { "react-jsx": 3 },
  "bs-dependencies": ["@rescript/react"]
}
```

enable JSX support →

add React support →

A SIMPLE REACT EXAMPLE

```
type buttonType = Primary | Link

module Button = {
  @react.component
  let make = (~variant=Primary, ~children) => {
    let className = switch(variant) {
      | Primary => "btn btn-primary"
      | Link => "btn btn-link"
    }

    <button className>{children}</button>
  }
}
```

```
// Unfortunately ReScript's bindings are not yet updated for 18.0.0
// but we can easily fix this ourselves.
type reactRoot
@module("react-dom/client")
external createRoot : Dom.element => reactRoot = "createRoot"
@send
external render : (reactRoot, React.element) => unit = "render"

// Tell ReScript about JavaScript's `document.body` property.
@val @scope("document") external body : Dom.element = "body"
```

```
body
  ->createRoot
  ->render(
    <div>
      <Button>{React.string("Continue")}</Button>
      <Button variant={Link}>{React.string("cancel")}</Button>
    </div>
  )
```

A BUTTON COMPONENT

type definition

module declaration

function

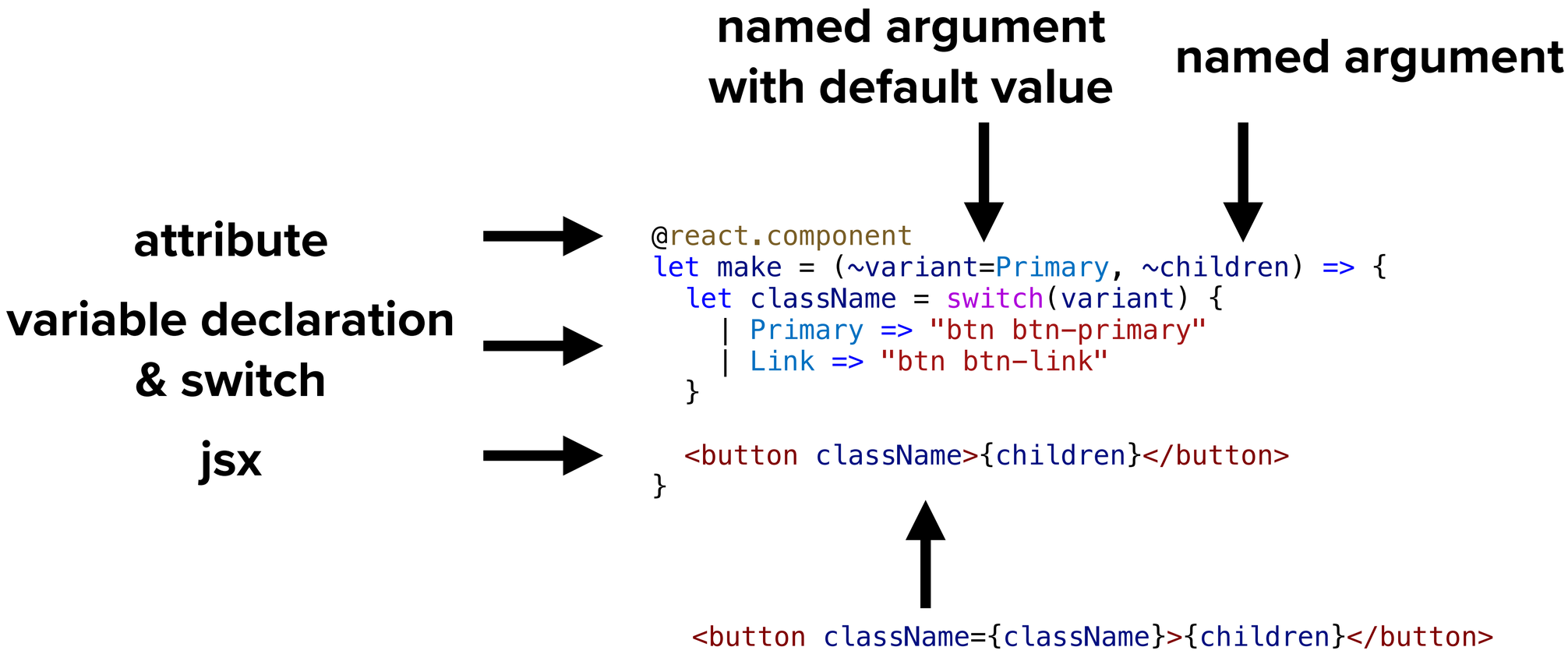
variant

```
type buttonType = Primary | Link

module Button = {
  @react.component
  let make = (~variant=Primary, ~children) => {
    let className = switch(variant) {
      | Primary => "btn btn-primary"
      | Link => "btn btn-link"
    }

    <button className>{children}</button>
  }
}
```

A BUTTON COMPONENT



ACCESSING EXTERNAL JAVASCRIPT

type definition

binding

binding

binding

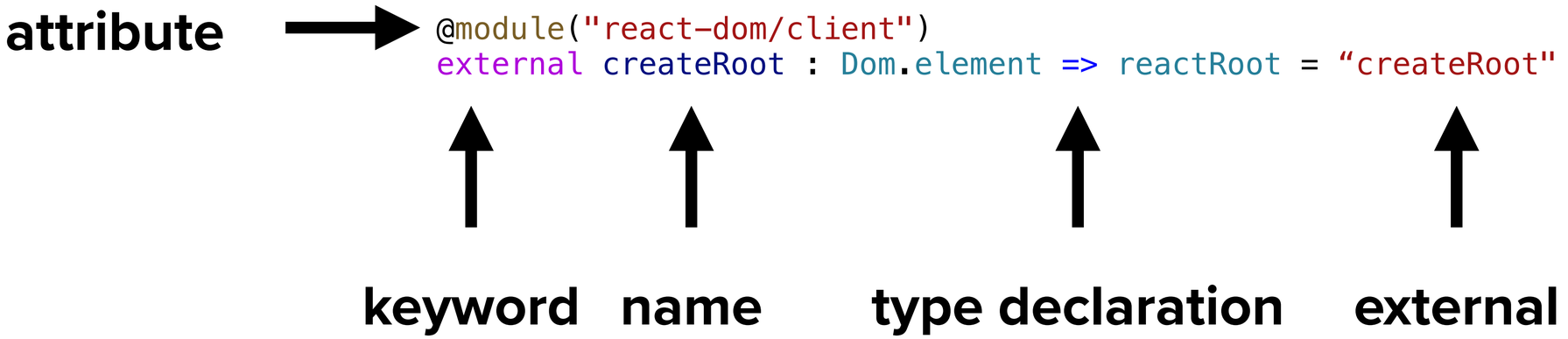
```
// Unfortunately ReScript's bindings are not yet updated for 18.0.0
// but we can easily fix this ourselves.
type reactRoot

@module("react-dom/client")
external createRoot : Dom.element => reactRoot = “createRoot"

@send
external render : (reactRoot, React.element) => unit = "render"

// Tell ReScript about JavaScript's `document.body` property.
@val
@scope(“document")
external body : Dom.element = "body"
```

ACCESSING EXTERNAL JAVASCRIPT




```
import * as Client from "react-dom/client";
Client.createRoot(/* Dom.element */)
```

ACCESSING EXTERNAL JAVASCRIPT

structure

```
@optionalAttributes
external rescriptName : type => "externalName"
```

```
@module("react-dom/client")
external createRoot : Dom.element => reactRoot = “createRoot"

@send
external render : (reactRoot, React.element) => unit = "render"

@val
@scope(“document")
external body : Dom.element = "body"
```

RENDERING OUR APPLICATION

```
body
  ->createRoot
  ->render(
    <div>
      <Button>{React.string("Continue")}</Button>
      <Button variant={Link}>{React.string("cancel")}</Button>
    </div>
  )
```

```
render(
  createRoot(body),
  <div>
    <Button>{React.string("Continue")}</Button>
    <Button variant={Link}>{React.string("cancel")}</Button>
  </div>
)
```

A SIMPLE REACT EXAMPLE

```
type buttonType = Primary | Link

module Button = {
  @react.component
  let make = (~variant=Primary, ~children) => {
    let className = switch(variant) {
      | Primary => "btn btn-primary"
      | Link => "btn btn-link"
    }

    <button className>{children}</button>
  }
}
```

```
// Unfortunately ReScript's bindings are not yet updated for 18.0.0
// but we can easily fix this ourselves.
type reactRoot
@module("react-dom/client")
external createRoot : Dom.element => reactRoot = "createRoot"
@send
external render : (reactRoot, React.element) => unit = "render"

// Tell ReScript about JavaScript's `document.body` property.
@val @scope("document") external body : Dom.element = "body"

body
  ->createRoot
  ->render(
    <div>
      <Button>{React.string("Continue")}</Button>
      <Button variant={Link}>{React.string("cancel")}</Button>
    </div>
  )
```

A SIMPLE REACT EXAMPLE

```
// Generated by ReScript, PLEASE EDIT WITH CARE

import * as React from "react";
import * as Client from "react-dom/client";

function ReactExample$Button(Props) {
  var variantOpt = Props.variant;
  var children = Props.children;
  var variant = variantOpt !== undefined
                  ? variantOpt
                  : /* Primary */0;
  var className = variant
                  ? "btn btn-link"
                  : "btn btn-primary";
  return React.createElement(
           "button",
           { className: className },
           children
  );
}

var Button = {
  make: ReactExample$Button
};
```

```
Client.createRoot(document.body)
      .render(
        React.createElement(
          "div",
          undefined,
          React.createElement(
            ReactExample$Button,
            { children: “Continue" }
          ),
          React.createElement(
            ReactExample$Button,
            {
              variant: /* Link */1,
              children: "cancel"
            }
          )
        )
      );

export {
  Button ,

}
/*  Not a pure module */
```

A SIMPLE REACT EXAMPLE

```
// Generated by ReScript, PLEASE EDIT WITH CARE

import * as React from "react";
import * as Client from "react-dom/client";

function ReactExample$Button(Props) {
  var variantOpt = Props.variant;
  var children = Props.children;
  var variant = variantOpt !== undefined
                ? variantOpt
                : /* Primary */0;
  var className = variant
                ? "btn btn-link"
                : "btn btn-primary";
  return React.createElement(
           "button",
           { className: className },
           children
  );
}
```

```
Client.createRoot(document.body)
      .render(
        React.createElement(
          "div",
          undefined,
          React.createElement(
            ReactExample$Button,
            { children: “Continue" }
          ),
          React.createElement(
            ReactExample$Button,
            {
              variant: /* Link */1,
              children: "cancel"
            }
          )
        )
      );

export {


}
/*  Not a pure module */
```

TRY TO BREAK IT

TRY TO BREAK IT

```
type buttonType = Primary | Secondary | Link
```

```
Warning number 8
rescript-presentation/src/ReactExample.res:6:21-9:5

 4 | @react.component
 5 | let make = (~variant=Primary, ~children) => {
 6 |   let className = switch(variant) {
 7 |     | Primary => "btn btn-primary"
 8 |     | Link => "btn btn-link"
 9 |   }
10 |
11 |   <button className>{children}</button>

You forgot to handle a possible case here, for example:
Secondary
```

bsconfig.json

```
"warnings": {
  "error": "+8"
}
```

TRY TO BREAK IT

```
type buttonType = Primary | Secondary | Link
```

```
Warning number 8 (configured as error)
rescript-presentation/src/ReactExample.res:6:21-9:5

 4 | @react.component
 5 | let make = (~variant=Primary, ~children) => {
 6 |   let className = switch(variant) {
 7 |     | Primary => "btn btn-primary"
 8 |     | Link => "btn btn-link"
 9 |   }
10 |
11 |   <button className>{children}</button>

You forgot to handle a possible case here, for example:
Secondary
```

bsconfig.json

```
"warnings": {
  "error": "+8"
}
```

PROVIDED VARIANTS

Option

```
type option<'some> = Some('some) | None

switch (maybeValue) {
  | Some(actualValue) => doSomething(actualValue)
  | None => handleMissing()
}
```

Result

```
type result<'ok, 'error> = Ok('ok) | Error('error)

switch (readFile()) {
  | Ok(fileContents) => doSomethingWith(fileContents)
  | Error(readError) => showErrorToUser(readError)
}
```

TRY TO BREAK IT

```
let className = switch(variant) {
  | Primary => "btn btn-primary"
  | Link => "btn btn-link"
  | NonExistent => "btn"
}
```

```
  We've found a bug for you!
  rescript-presentation/src/ReactExample.res:9:9-19

   7 |     | Primary => "btn btn-primary"
   8 |     | Link => "btn btn-link"
   9 |     | NonExistant => "btn"
  10 |   }
  11 |

  This variant pattern is expected to have type buttonType
  The constructor NonExistant does not belong to type buttonType

FAILED: cannot make progress due to previous errors.
```

TRY TO BREAK IT

```
body
  ->render(
    <div>
      <Button>{React.string("Continue")}</Button>
      <Button variant={Link}>{React.string("cancel")}</Button>
    </div>
  )
```

```
We've found a bug for you!
rescript-presentation/src/ReactExample.res:26:1-4

24 | @val @scope("document") external body : Dom.element = "body"
25 |
26 | body
27 |   ->render(
28 |     <div>

This has type:
  Dom.element (defined as
    Dom.eventTarget_like<Dom._node<Dom._element<Dom._baseClass>>>)
Somewhere wanted: reactRoot

FAILED: cannot make progress due to previous errors.
```

TRY TO BREAK IT

```
<Button variant={Link}></Button>
```

```
We've found a bug for you!
rescript-presentation/src/ReactExample.res:31:8-13

29 ┆   <div>
30 ┆     <Button>{React.string("Continue")}</Button>
31 ┆     <Button variant={Link}></Button>
32 ┆   </div>
33 ┆ )

This has type:
  (~children: 'a) => {"children": 'a, "variant": option<buttonType>}
Somewhere wanted:
  {"children": React.element, "variant": option<buttonType>}

FAILED: cannot make progress due to previous errors.
```

Original Photo by Sumeet B on Unsplash

WHAT IS THE COMMUNITY WORKING ON?

- ReScript 10/10.1
  - async/await syntactic sugar (merged)
  - Improved JSX support with prop-spread

- ReScript Relay Router
  - Improve on an already great GraphQL integration
  - Smart per-page data loading on top of React Router
  - Flat waterfalls thanks to Suspense and SSR with client side continuation

WHERE CAN I GO FOR MORE?

Documentation

https://rescript-lang.org/

Community

https://forum.rescript-lang.org/

Me

https://www.alexandervarwijk.com/

RESCRIPT