Type Definition

Let's learn how to define our own types!

WE'LL COVER THE FOLLOWING

In the previous lessons, we learned how to define named identifiers. A cool feature of ReasonML is that it allows us to name data types as well.

Just as we used `let` for variables, we must use the `type` keyword to specify our own type.

Custom Names for Basic Types

Let’s try to create an `age` type which is simply an integer:

```
type age = int;
```

We can now use this `age` type in our `let` bindings:

```
type age = int;
let twoYears: age = 2;
Js.log(twoYears); /* 2 */
```

For the Sake of Documentation

Type definitions may seem trivial. After all, `age` is just an integer. The role of type definitions is to help us document our code according to the purpose of the program.

In several cases, it may be simpler to deal with the `age` type instead of calling it an integer. Hence, we can say that defining our own types can make the code more readable and suitable for use.

In the final lesson of this section, we will learn that there is a certain way to make `let` bindings mutable.