Creating generic type aliases

We can use generics in type aliases as well as interfaces. In this lesson, we will learn how to do this.

WE'LL COVER THE FOLLOWING

Generic type alias syntax

We can pass types into a type alias using the following syntax:

```
type TypeName<T1, T2, ...> = {
  ...
}
```

The members of the type can reference the generic types passed into it.

Generic type alias example

We can use a type alias to build the generic form we built using an interface in the last lesson. Try to do this in the code widget below. The `Contact` interface and `contactForm` variable remain the same, so they have already been implemented.

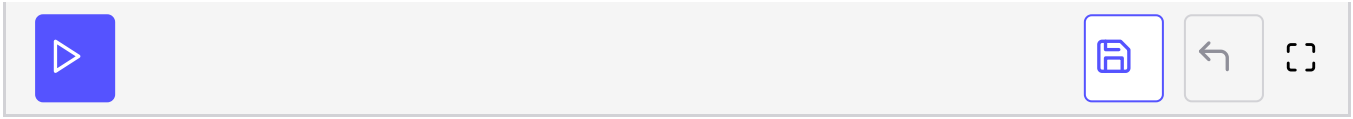
</> TypeScript

```
// TODO - create the Form generic type

interface Contact {
  name: string;
  email: string;
}

const contactForm: Form<Contact> = {
  errors: {
```

```
    email: "This must be a valid email address"
  },
  values: {
     name: "Bob",
     email: "bob@someemail.com"
   }
};
```



Show Answer

Wrap up

Generic type aliases are just like generic interfaces with a slightly different syntax. Generally, it is a personal preference which approach you use to create generic types.

Great stuff, we are getting comfortable with generics now!

In the next lesson, we'll learn all about generic classes.