

Manipulating Prototypes

getting and setting prototypes of objects using the Reflect API

We can get the prototype of an object using the Reflect API.

```
let classOfMyAccount = Reflect.getPrototypeOf( myAccount );  
  
console.log( classOfMyAccount.prototype === myAccount.prototype );
```



As you can see from the example, this prototype is the same as the prototype of our `PrivateAccount` class.

We can also set prototypes using `Reflect.setPrototypeOf`:

```
let newProto = {  
  get contact() {  
    return `${this.name} - 555-1269`;  
  }  
}  
  
Reflect.setPrototypeOf( myAccount, newProto );  
  
console.log( myAccount.contact );  
//> "Zsolt - 555-1269"
```



In the above example, we set the prototype of `myAccount` to `newProto`, and define a `contact` getter function. After changing the prototype, the new getter method is executed.

In the next lesson, we will discuss how we can access properties of objects and modify them using Reflection.

