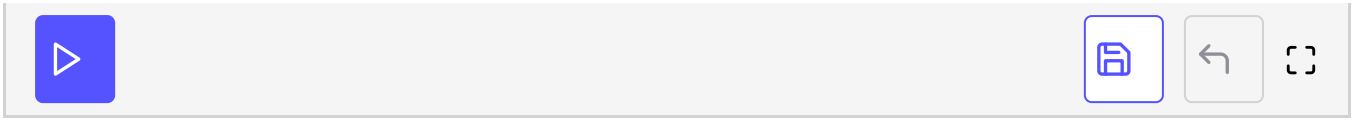
Introduction to TypeScript's Many Objects

In this lesson, we will introduce the many different types of objects in TypeScript.

TypeScript has many different object types.

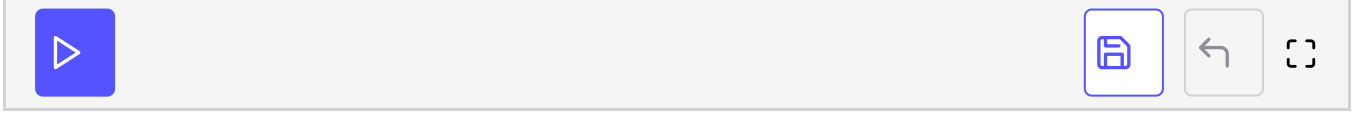
In this section, we will see the difference between the two objects which are confusingly called `Object` and `object` – one that starts with an uppercase `O` and the other one with a lowercase `o`.

```
let o1: Object = "I am an Object";
let o2: object = { text: "I am an object" };
let o3: Object = { text: "I am also an Object" }
let o4: object = ["I", "am", "an", "object"];
let o5: Object = ["I", "am", "an", "Object"];

console.log(o1);
// console.log(o2.text); // Does not compile
// console.log(o3.text); // Does not compile
console.log(o4);
console.log(o5);
```



We will also see how the curly bracket brings a third object type.

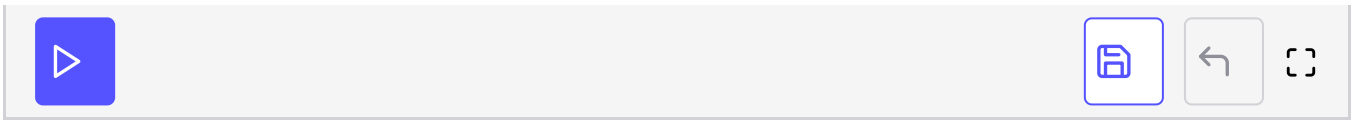
```
let curly1 = { text: "An object" };
let curly2: { text: string } = { text: "An object" };
let curly3: {} = { text: "An object" };

console.log(curly1.text);
console.log(curly2.text);
// console.log(curly3.text); // Does not not compile
```



The lowercase object comes not only from `Object.create` but also if you instantiate a class in TypeScript. It represents all types that are not primitive, which means those that are not `Boolean`, `number`, `string`, `symbol`, `null` or

`undefined`.

```
let create1 = Object.create({ text: "I am a created object" });
let p = { text: "I am an object" };
let create2 = Object.create(p);

console.log(create1.text);
console.log(p.text);
console.log(create2.text);
```



In the upcoming lesson, we'll study the curly braces object in more detail.