

Creating generic classes

We are going to learn how to create a generic class in this lesson.

WE'LL COVER THE FOLLOWING ^

- Generic class syntax
- Generic class example
- Wrap up

Generic class syntax

The syntax for a generic class is similar to that of a generic interface:

```
class ClassName<T1, T2, ...> {  
    ...  
}
```

The members of the class can reference the types passed into it.

Generic class example

Let's implement a generic list class where the type for the list items is passed into the class as a generic parameter. The start of our implementation is below:

</> TypeScript

```
class List<ItemType> {  
    private items: ItemType[] = [];  
}
```



Continue the implementation by adding a strongly typed `add` method to the

Continue the implementation by adding a strongly-typed `add` method to the class. It should have a single parameter for the list item to be added. The implementation should add the new list item to the `items` array.

 Show Answer

Create an instance of this generic class below the class definition in the code widget above:

```
const numberList = new List<number>();
```

So, we have created a class that will manage a list of numbers.

Add a number to `numberList`:

```
numberList.add(1);
```

Try adding a string to `numberList`:

```
numberList.add("2");
```

Do we receive a type error?

 Show Answer

Wrap up

Using generic classes allows classes to be created for general situations that can be applied to a specific situation by supplying types as parameters.

More information can be found on generic classes in the [TypeScript handbook](#).

Well done, we are now comfortable creating our own generic types!

Next up is a quiz to test what we have learned.

