Creating Abstract Classes

introduction to an abstract class and its implementation using a stock trading example

Abstract classes are classes that cannot be instantiated. Recall the `Shape` class in the previous example. Until we know what kind of shape we are talking about, we cannot do much with a generic shape.

Often, you have a couple of business objects on the same level. Assuming that you are not in the WET (We Enjoy Typing) group of developers, it is natural that you abstract the common functionalities into a base class. For instance, in the case of stock trading, you may have a `BarChartView`, a `LineChartView`, and a `CandlestickChartView`. The common functionalities related to these three views are abstracted into a `ChartView`. If you want to make `ChartView` abstract, do the following:

```
class ChartView {
    constructor( /* ... */ ) {
        if ( new.target === ChartView ) {
            throw new Error(
                'Abstract class ChartView cannot be instantiated.' );
        }
        // ...
    }
    // ...
}
```

The built-in property `new.target` contains a reference to the class written next to the `new` keyword during instantiation. This is the class whose constructor was first called in the inheritance chain.

In the next lesson, let's talk about getters and setters.