

## 17.16 Lambda Event Handlers

In [Section 12.5.5](#), you learned how to implement an event handler using an anonymous inner class. Event-listener interfaces with one `abstract` method—like `ChangeListener`—are functional interfaces. For such interfaces, you can implement event handlers with lambdas. For example, the following `Slider` event handler from [Fig. 12.23](#):

---

```
tipPercentageSlider.valueProperty().addListener(  
    new ChangeListener<Number>() {  
        @Override  
        public void changed(ObservableValue<? extends Number> oldValue, Number newValue) {  
            tipPercentage =  
                BigDecimal.valueOf(newValue.intValue() / 100);  
            tipPercentageLabel.setText(percent.format(tipPercentage));  
        }  
    }  
);
```

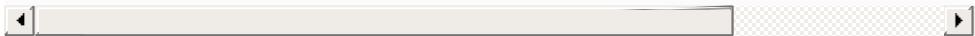


can be implemented more concisely with a lambda as

---

```
tipPercentageSlider.valueProperty().addListener(  
    (ov, oldValue, newValue) -> {  
        tipPercentage =  
            BigDecimal.valueOf(newValue.intValue() / 100);  
        tipPercentageLabel.setText(percent.format(tipPercentage));  
    }  
>
```

```
    tipPercentageLabel.setText(percent.format(tipPe
}));
```



For a simple event handler, a lambda significantly reduces the amount of code you need to write.