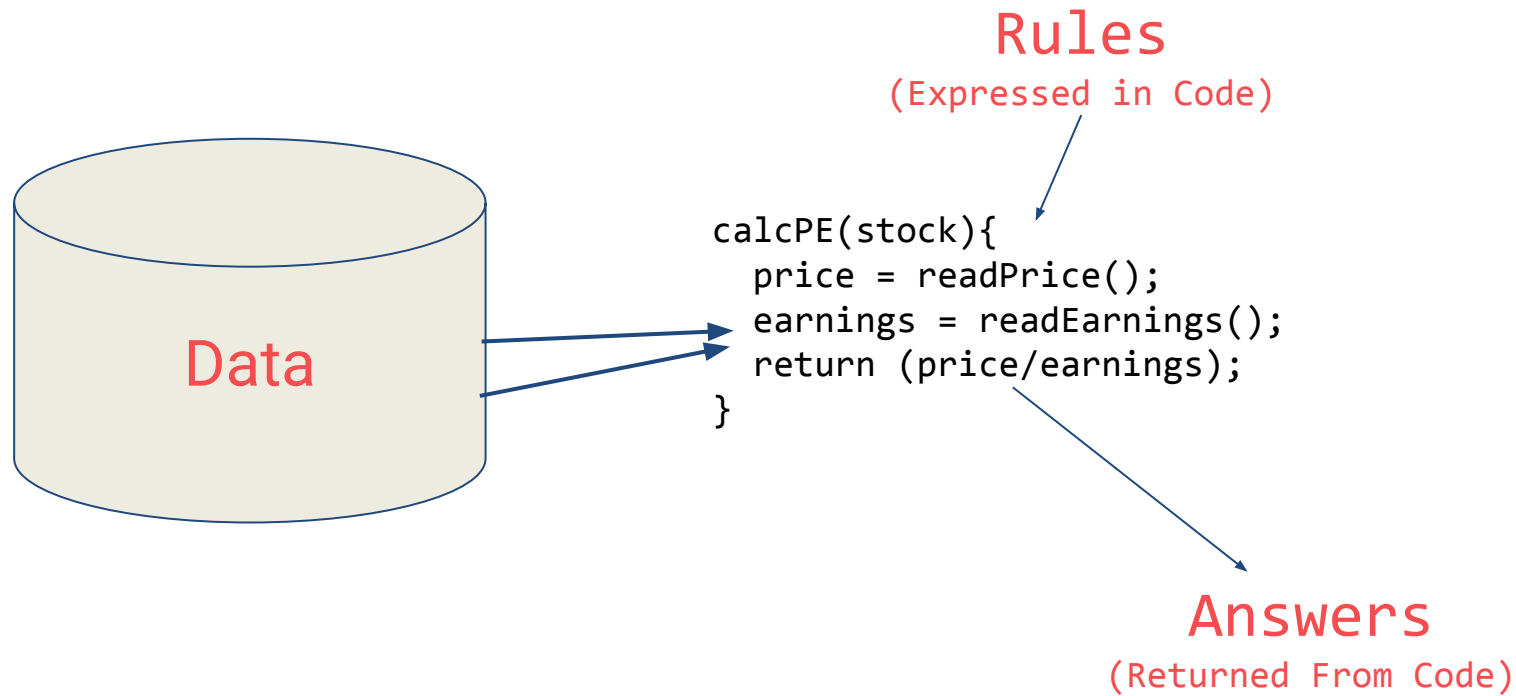


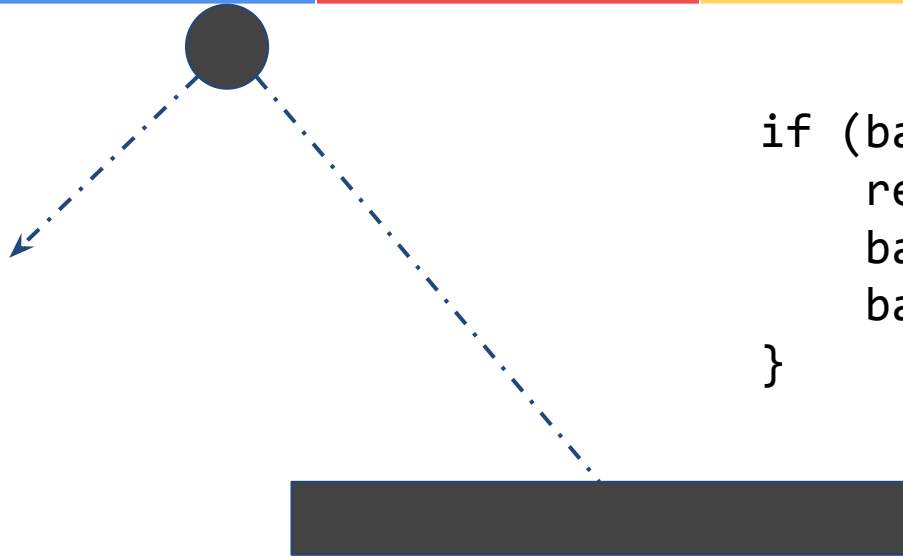
Copyright Notice

These slides are distributed under the Creative Commons License.

[DeepLearning.AI](#) makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite [DeepLearning.AI](#) as the source of the slides.

For the rest of the details of the license, see <https://creativecommons.org/licenses/by-sa/2.0/legalcode>





```
if (ball.collide(brick)){  
    removeBrick();  
    ball.dx=-1*(ball.dx);  
    ball.dy=-1*(ball.dy);  
}
```





Activity Recognition



```
if(speed<4){  
    status=WALKING;  
}
```

Activity Recognition



```
if(speed<4){  
    status=WALKING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}
```



Activity Recognition



```
if(speed<4){  
    status=WALKING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else if(speed<12){  
    status=RUNNING;  
} else {  
    status=BIKING;  
}
```



Activity Recognition



```
if(speed<4){  
    status=WALKING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else if(speed<12){  
    status=RUNNING;  
} else {  
    status=BIKING;  
}
```



// Oh crap



Activity Recognition



0101001010100101010
1001010101001011101
0100101010010101001
0101001010100101010

Label = WALKING



1010100101001010101
0101010010010010001
0010011111010101111
1010100100111101011

Label = RUNNING



10010100111111010101
1101010111010101110
1010101111010101011
1111110001111010101

Label = BIKING



11111111111010011101
0011111010111110101
0101110101010101110
1010101010100111110

Label = GOLFING
(Sort of)

$X = -1, 0, 1, 2, 3, 4$

$Y = -3, -1, 1, 3, 5, 7$



```
model = tf.keras.Sequential([  
    tf.keras.Input(shape=(1,)),  
    tf.keras.layers.Dense(units=1)  
])
```



```
model = tf.keras.Sequential([  
    tf.keras.Input(shape=(1,)),  
    tf.keras.layers.Dense(units=1)  
])
```



```
model = tf.keras.Sequential([  
    tf.keras.Input(shape=(1,)),  
    tf.keras.layers.Dense(units=1)  
])
```



```
model = tf.keras.Sequential([  
    tf.keras.Input(shape=(1,)),  
    tf.keras.layers.Dense(units=1)  
])
```




```
model = tf.keras.Sequential([  
    tf.keras.Input(shape=(1,)),  
    tf.keras.layers.Dense(units=1)  
)  
model.compile(optimizer='sgd', loss='mean_squared_error')
```

```
model = tf.keras.Sequential([
    tf.keras.Input(shape=(1,)),
    tf.keras.layers.Dense(units=1)
])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)
```



```
model = tf.keras.Sequential([
    tf.keras.Input(shape=(1,)),
    tf.keras.layers.Dense(units=1)
])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)
```



```
model = tf.keras.Sequential([
    tf.keras.Input(shape=(1,)),
    tf.keras.layers.Dense(units=1)
])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)

model.predict(np.array([10.0]))
```

