**Training Days**

As a seasoned athlete, one of your favorite activities is running marathons. You use a service called Training Days that sends you a message for the event you signed up for and the days you have left to train.

Since you also code, Training Days has asked you to help them solve a problem: The program currently uses the wrong scope for its variables. They know this can be troublesome as their service evolves. In this project you will make Training Days more maintainable and less error-prone by fixing variable scopes.

**Code:**

**// The scope of `random` is too loose**

**const getRandEvent = () => {**

**const random = Math.floor(Math.random() \* 3);**

**if (random === 0) {**

**return 'Marathon';**

**} else if (random === 1) {**

**return 'Triathlon';**

**} else if (random === 2) {**

**return 'Pentathlon';**

**}**

**};**

**// The scope of `days` is too tight**

**const getTrainingDays = event => {**

**let days = 50**

**if (event === 'Marathon') {**

**days = 50;**

**} else if (event === 'Triathlon') {**

**days = 100;**

**} else if (event === 'Pentathlon') {**

**days = 200;**

**}**

**return days;**

**};**

**let name = 'Nala';**

**// The scope of `name` is too tight**

**const logEvent = (name,event) => {**

**console.log(`${name}'s event is: ${event}`);**

**};**

**const logTime = (name,days) => {**

**console.log(`${name}'s time to train is: ${days} days`);**

**};**

**const event = getRandEvent();**

**const days = getTrainingDays(event);**

**// Define a `name` variable. Use it as an argument after updating logEvent and logTime**

**logEvent(event);**

**logTime(days);**

**const event2 = getRandEvent();**

**const days2 = getTrainingDays(event2);**

**const name2 = 'Warren';**

**logEvent(name2, event2);**

**logTime(name2, days2);**