

# Quiz Game

Let's create a quiz game in JavaScript!

This is about as simple as it sounds, let's create a command line game in Node that asks the user questions, and allows the user to answer.

## First Iteration

Create the basic quiz. Create a "question" class. Each question should have the text for the question (obviously), an answer, and an ID in the form of an integer. Create the Quiz class to move the user from question to question. Quiz the user with the questions, and if they provide the correct answer allow them to move on to the next question. If the user incorrectly answers, notify the user and repeat the same question again.

With Ruby, we could easily get user input through `gets.chomp`. Node is different, but luckily we have a pretty nice library in "**read**", brought to you by npm.

So `npm install read` in your exercise folder and let's get rolling with an example.

Our first step is to require the read library in our JavaScript file:

```
var read = require('read');
```

Read requires two things to work, options for the prompt in the form of an object literal, and a callback function to be executed whenever the user enters input:

```
options = {
  prompt: "What's your name?\n>"
}
// Our options object, the prompt is simply what will appear in the command line when read is called

read(options, displayName)
// The prompt itself, passing options, and using our callback function after input

function displayName (err, name){
  console.log("Your name is: " + name)
}
// Outputs whatever the user has entered back to the console
```

That is the simple example of how read works, keep in mind that the options object can be manipulated just like a hash. A list of more options can be found at (npm)

[\[https://www.npmjs.com/package/read\]](https://www.npmjs.com/package/read)

## Second Iteration

Add a new field to the Question class, with a point value. If the user answers a question correctly, add to a new “total” variable. Keep track of the total points accumulated throughout the quiz, and display it to the user while asking the question. If a user incorrectly answers the question, make sure to subtract the quizzes point value from the total before repeating.

## Third Iteration

Bonus Question Randomly assign a question to be a bonus question for each quiz. Make this random question worth double the points, and display to the user that they correctly answered the bonus question correctly if they do.

**Hint:** for the bonus question try out `Math.random()`. `Math.random()` returns a random number between 0 and 1. So in order to get a random whole number we need to do something like this:

```
var randomnumber = Math.floor(Math.random()*11)
console.log(randomnumber)
// This will output a random number between 1 and 10
```

You can manipulate the number by multiplying by a number other than 10, ie: if you multiply by 16 you will generate a random number between 1 and 15.

In order to select a random question as the bonus question, matchup this knowledge with our “id” attribute that our question class, select one, and manipulate it’s point value before the quiz begins.

## Fourth Iteration

Let’s create a User class now. Keep it simple, the only attributes our user should have are a name, their current points total, and the current question number they are on. Hard Code the user into the application, for now. **\*\*Hint:** use the question ID as a marker for what question the User is currently on. Instead of immediately opening the quiz when you load your application, prompt the user with “New User?”. If the answer is yes, allow them to “register”. If the answer is no, then prompt them again with a “login”. Accept the user’s name as their login, if it matches an existing user, take them to the quiz.

## Fifth Iteration

Let's save our users in a file, so we can save our game state as well. When somebody types "save" in during the quiz save the User's state in a JSON file. Now, when we login as a user, take them to the question number that they were on when they typed "save". Make sure their point total is also persisted from your save file.

# Challenge Mode

Show the user a leaderboard on load in ASCII. For example:

```
Josh ----- 10 Points  
Nizar ----- 6 Points  
Fernando ----- 3 Points
```

If you pass the current leader during your game, notify the user in the console.