Exercise 11: Function Documentation 2

For this exercise, you will document some more functions using the best practices I gave in the previous lecture. If no return value is mentioned in the notes (it doesn't return or get anything), then the function has no return value.

Example You Can Use as a Template

As a reminder, here is an example you can use as a template for the overall structure.

Documentation:

getFriendDistance(username)

Returns the "degrees of separation" between the specified user and the current user.

Any user is some "degrees of separation" from another user. For example, if they are friends, the distance is one, and if they are friends of friends, the distance is two. This function returns the distance from the specified user to the current user. If there is no connection between the users, then it returns NaN.

Parameters:

- **username**
  Type: String
  Username of the user to get the distance from

Returns:

Type: Number
Number of degrees of separation between the users

Information from the Developer Team

- **safeMode(enable)**

Enables or disables safe mode, depending on the parameter that is passed in. No return value.

- **getPremiumMode()**

Gets whether the user has paid for premium mode.

- **setTimeout(timeout)**

Sets the time to wait for a response before timing out. The parameter value is in milliseconds.

- **newEvent(calendarId)**

Creates a new event for the specified calendar with the ID passed in as the parameter. Calendar ID is a number. Returns the ID for the new event, which will be a number.

- **getDirections(latitude, longitude, directionsCallback)**

Latitude and longitude are numbers (degrees) for the destination location. directionsCallback is a function that is called when the directions come back from the serve r. It has a string parameter called directions which contains the directions on how to get from the current location to the destination location.
(Hint: use bold to refer to any function and variable names.)

- **createTransaction(amount)**

Creates a new transaction, where the amount parameter is a number in the default currency. Returns the new transaction ID, which is a String. Other functions that developers might be interested in are getTransaction and deleteTransaction.

Take a Look at How I've Done It

Remember, there's no one right way to do this, but here's how I would do it: http://sdkbridge.com/prog1/Exercise11Answers.pdf.