JavaScript Sleep Calculator Instructions

In this activity, you will calculate if you're getting enough sleep each week using a sleep calculator.

The program will determine the actual and ideal hours of sleep for each night of the last week.

Finally, it will calculate, in hours, how far you are from your weekly sleep goal.

1	The first problem to solve is determining how many hours of sleep you got each night of the week. You can create a function that returns any given night's number of hours of rest. Instead of writing seven different functions (one for each day of the week), let's write one function with a parameter for the day. Create a function named getSleepHours with a single parameter named day.

Hint	For a function with a single parameter, the syntax looks like this:
	
	``js
	const exampleName = exampleParameter => {
	
	};
	``

2	The function should accept a day as an argument and return the number of hours you slept that night. For instance, if you got 8 hours of sleep on Monday night, calling getSleepHours('monday') should return 8. Use an if/else or switch statement to implement this.

Hint	```js
	const getSleepHours = day => {
	  if (day === 'monday') {
	    return 8;
	  } else if {
	    // continue else if's...
	  }
	};
	``

3	Test the function by calling it multiple times and printing the results to the console. You can remove the tests when you know your function works.

Hint	```js
	console.log(getSleepHours('tuesday')); // should print the # hours assigned to tuesday
	``

4	Now that you've written a function to get the sleep hours for each night, you need to do three things: • Get the total sleep hours that you actually slept • Get the ideal sleep hours that you prefer • Calculate the sleep debt, if any. To get the total sleep hours that you actually slept, create a new function named getActualSleepHours that takes no parameters.

Hint	```const getSleepHours = day => {```
| | ```javascript
const getSleepHours = day => {

  ...
};


const getActualSleepHours = () => {

  ...
};
``` |
| 5 | Inside the getActualSleepHours() function, call the getSleepHours() function for each day of the week. Add the results together and return the sum using an implicit return. |
| Hint | The concise body form of a function uses arrow notation and does not include brackets or the return keyword.<br><br>```javascript
const getActualSleepHours = () => getSleepHours('Monday') + getSleepHours('Tuesday') +
getSleepHours('Wednesday') +
...;
``` |
| 6 | To get the ideal sleep hours that you prefer, create a function named getIdealSleepHours with no parameters.

Inside the function, declare a variable named idealHours and set its value to your ideal hours per night. Then return the idealHours multiplied by 7.

You'll want to multiply by 7 to get the total hours you prefer per week. |
| Hint | ```javascript
const getIdealSleepHours = () => {
  const idealHours = 7.5;
  return idealHours * 7;
};
``` |
| 7 | Test your two new functions by calling them and printing the results to the console.

You can remove the tests when you know your functions works. |
| Hint | ```javascript
console.log(getActualSleepHours()); // should print the sum of all sleep hours in the week

console.log(getIdealSleepHours()); // if idealHours is 8, should print 56
``` |
| 8 | Now that you can get the actual sleep hours and the ideal sleep hours, it's time to calculate sleep debt.

Create a function named calculateSleepDebt with no parameters.

Inside of its block, create a variable named actualSleepHours set equal to the getActualSleepHours() function call.

Then, create another variable named idealSleepHours, set equal to the getIdealSleepHours() function call. |
| Hint | ```javascript
const calculateSleepDebt = () => {
  const actualSleepHours = getActualSleepHours();
  const idealSleepHours = getIdealSleepHours();
};
``` |
| 9 | Now that you have actualSleepHours and idealSleepHours, you can write a few if/else statements to output the result to the console. The function should fulfill this logic:

• If actual sleep equals ideal sleep, log to the console that the user got the perfect amount of sleep. |

	• If the actual sleep is greater than the ideal sleep, log to the console that the user got more sleep than needed. • If the actual sleep is less than the ideal sleep, log to the console that the user should get some rest.
Hint	actualSleepHours must be equal to, greater than, or less than idealSleepHours. The if and else if statements cover the first two cases, and the else statement covers the 'less than' case. ```js\nif (actualSleepHours === idealSleepHours) {\n console.log('...');\n} else if (actualSleepHours > idealSleepHours) {\n console.log('...');\n} else {\n console.log('...');\n}\n```
10	To make this calculator more helpful, add the hours the user is over or under their ideal sleep in each log statement in calculateSleepDebt().
Hint	You can interpolate the maths inside the string passed to console.log() to print. For instance, if the user got less sleep than is ideal, you could write: ```js\nif (actualSleepHours < idealSleepHours) {\n console.log('You got ' + (idealSleepHours - actualSleepHours) + ' hour(s) less sleep than you needed this week. Get some rest.');\n}\n```
11	On the last line of the program, start the program by calling the calculateSleepDebt() function.
Hint	```js\ncalculateSleepDebt();\n```
12	Stretch & Challenge • getActualSleepHours() could be implemented without calling getSleepHours(). Use literal numbers and the + operator to rewrite getActualSleepHours(). It should still return the total actual hours slept in the week. • Some people need to sleep longer than others. Rewrite getIdealSleepHours() so that you can pass it an argument, like getIdealSleepHours(8) where 8 is the ideal hours per night. Update the call to getIdealSleepHours() in calculateSleepDebt() too. To see the solutions, open the hint.

Hint	Put the daily sleep hours directly into getActualSleepHours().
	
	```js\nconst getActualSleepHours = () => 6 + 7 + 9 + 8 + 5 + 10 + 11;\n```
	Make idealHours a parameter and multiply it by 7.
	
	```js\nconst getIdealSleepHours = idealHours => idealHours * 7;\n```
	When you call the updated function in calculateSleepDebt(), call it like this:
	
	```js\nconst calculateSleepDebt = () => {\n\n ...\n\n const idealSleepHours = getIdealSleepHours(8);\n\n ...\n};\n```