**Study Time Calculator**

When you're caught in a busy lifestyle, it’s easy to lose track of how much time you have to spare. Study Time Calculator is a simple tool that helps you calculate the hours you have available in a given week, and how you can fit online study into your lifestyle.

In this project we’ll calculate if you’re getting enough study time each week using a Study Time Calculator.

The program will determine the actual and ideal hours of study time for each day of the last week.

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

**1.**

The first problem to solve is determining how many hours of study you got each day of the week.

You can create a function that returns any given day’s number of hours of empty. 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 getStudyHours with a single parameter named day.

Hint

For a function with a single parameter, the syntax looks like this:

const exampleName = exampleParameter => {

};

**2.**

The function should accept a day as an argument and return the number of hours you studied that day.

For instance, if you got 8 hours of study on Monday, calling getStudyHours('monday') should return 8.

Use an if/else or switch statement to implement this.

Hint

const getStudyHours = 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

console.log(getStudyHours('tuesday')); // should print the # hours assigned to tuesday

**4.**

Now that you’ve written a function to get the study hours for each day, we need to do three things:

* Get the total study hours that you actually study
* Get the ideal study hours that you prefer
* Calculate the study debt, if any.

To get the total study hours that you actually study, create a new function named getActualStudyHours that takes no parameters.

Hint

const getStudyHours = day => {

...

};

const getActualStudyHours = () => {

...

};

**5.**

Inside the getActualStudyHours() function, call the getStudyHours() 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.

const getActualStudyHours = () => getStudyHours('Monday') + getStudyHours('Tuesday') + getStudyHours('Wednesday') +

...;

**6.**

To get the ideal study hours that you prefer, create a function named getIdealStudyHours with no parameters.

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

You’ll want to multiply by 7 to get the total hours you prefer per week.

Hint

const getIdealStudyHours = () => {

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

console.log(getActualStudyHours()); // should print the sum of all study hours in the week

console.log(getIdealStudyHours()); // if idealHours is 8, should print 56

**8.**

Now that you can get the actual study hours and the ideal study hours, it’s time to calculate study debt.

Create a function named calculateStudyDebt with no parameters.

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

Then, create another variable named idealStudyHours, set equal to the getIdealStudyHours() function call.

Hint

const calculateStudyDebt = () => {

const actualStudyHours = getActualStudyHours();

const idealStudyHours = getIdealStudyHours();

};

**9.**

Now that you have actualStudyHours and idealStudyHours, you can write a few if/else statements to output the result to the console. The function should fulfill this logic:

* If actual study equals ideal study, log to the console that the user got the perfect amount of study.
* If the actual study is greater than the ideal study, log to the console that the user got more study than planned. User can rest.
* If the actual study is less than the ideal study, log to the console that the user should create some empty time to study.

Hint

actualStudyHours must be equal to, greater than, or less than idealStudyHours. The if and else if statements cover the first two cases, and the else statement covers the ‘less than’ case.

if (actualStudyHours === idealStudyHours) {

console.log('...');

} else if (actualStudyHours > idealStudyHours) {

console.log('...');

} else {

console.log('...');

}

**10.**

To make this calculator more helpful, add the hours the user is over or under their ideal study in each log statement in calculateStudyDebt().

Hint

You can interpolate the math inside the string passed to console.log() to print. For instance, if the user got less study than is ideal, you could write:

if (actualStudyHours < idealStudyHours) {

console.log('You got ' + (idealStudyHours - actualStudyHours) + ' hour(s) less study than you needed this week. Create some empty time to study.');

}

**11.**

On the last line of the program, start the program by calling the calculateStudyDebt() function.

Hint

calculateStudyDebt();

**12.**

Good job. Your calculator is working. Now for extra practice, try these:

* getActualStudyHours() could be implemented without calling getStudyHours(). Use literal numbers and the + operator to rewrite getActualStudyHours(). It should still return the total actual hours study in the week.
* Some people can study longer than others. Rewrite getIdealStudyHours() so that you can pass it an argument, like getIdealStudyHours(8) where 8 is the ideal hours per day. Update the call to getIdealStudyHours() in calculateStudyDebt() too.

Hint

Put the daily study hours directly into getActualStudyHours().

const getActualStudyHours = () => 6 + 7 + 9 + 8 + 5 + 10 + 11;

Make idealHours a parameter and multiply it by 7.

const getIdealStudyHours = idealHours => idealHours \* 7;

When you call the updated function in calculateStudyDebt(), call it like this:

const calculateStudyDebt = () => {

...

const idealStudyHours = getIdealStudyHours(8);

...

};

**12.**

Additional Step! If you finish early do not pass the next Project, do this step.

**12.a.**

Make a new calculator (duplicate recent project) that uses different parameters not only days. Give every study time points.

#### Mornings

Our brains are the most switched-on in the first few hours of the day, and it’s well worth taking advantage of this when the chance arises. How many hours a week can you dedicate to studying in the morning? Multiply this time by 2,5.

#### Evenings

Fitting in as little as half an hour of study a night can pay dividends over the course of a week. How many hours a week can you dedicate to studying in the evening? Multiply this time by 1.

#### Commuting

Mobile technology has come a long way, and a long commute on public transport can be a blessing in disguise. How many hours a week can you dedicate to study while commuting to work? Multiply this time by 2.

#### Work

Why not use your lunch break for fitting in some online study? How many hours a week can you dedicate to studying at work, particularly during your lunch break? Multiply this time by 2.

#### Weekends

Weekends are a fantastic time to fit in some longer, more focused study sessions, without neglecting your family and social life. How many hours of study can you fit into your average weekend? Multiply this time by 3.

**12.b.**

Create a variable idealStudyPoints that keeps your goal. Under 25 points is low, 25 to 35 is enough, 35 to 45 is good, above 45 you are a superhero. Choose one.

**12.c.**

Now that you have actualStudyPoints and idealStudyPoints, you can write a few if/else statements to output the result to the console. Look at the step-9. Change study hours to calculate different goals and print to the console.

Congragulations.