**Udemy Coding Challenge**

**Calculate Temperature Amplitude**

A company is building a smart home thermometer. The company needs to calculate the temperature amplitude from a range of data elements. We also need to ignore any values that are incorrect.

There are a number of ways that we can solve this problem. One method would be to create a new array and take values from our ‘temperatures’ array and put them into a new array. We could add arguments to make sure that only the correct data is implemented into our new array.

1. First of all we need to create our temperatures array. Then we need to create an array where we want to take correct values and store them.
2. *const* temperatures = [
3. 3,
4. -2,
5. -6,
6. -1,
7. "error",
8. 9,
9. 1000,
10. -101,
11. 13,
12. 17,
13. 15,
14. 14,
15. 9,
16. 5,
17. ];
18. *let* numberArray = [];

Here we have created our initial array ‘temperatures’ and then added a new array ‘numberArray[]’ where we will store our correct figures.

2. We now need to think about what kind of information we want to put into our new array. We are dealing with temperature. Temperature is a number so first we only want to accept number values. Secondly we don’t want to accept values that are ‘extreme’. We can set an argument that only accepts data between certain values. Here is how we can do it;

for (*let* i = 0; i < temperatures.length; i++) {

if (

typeof temperatures[i] === "number" &&

temperatures[i] <= 100 &&

temperatures[i] >= -100

) {

numberArray.push(temperatures[i]);

} else {

console.log(`These are the errors: ${temperatures[i]}`);

}

}

Here we have created a for loop. The for loop will start to iterate through our data in the temperatures[] array. We have set the starting point to be at index position 0 with ‘let i = 0’. We set the stopping point to be the length of the temperatures[] with ‘i < temperatures.length’ and finally we want to iterate by one position each time; ‘i++’.

We then need to specify the code we want to run in our for loop. One method we could use is an if statement.

if (typeof temperatures[i] === 'number' && temperatures[i] <= 100 && temperatures[i] >= -100')

This if statement assesses whether;

1. The data is a typeof ‘number’. We do not want Booleans for example.
2. The data is between -100 and 100. This way we can avoid extreme values.

Finally; if this produces a truthy value, we want to push the data into our new array.

If the statement is not true then we want to display the errors.

**Graphical user interface, text, application, table

Description automatically generated**Here we have displayed the error code. So we know that our if/else statement is working.

3. We now have data entering our new array; numberArray[]. But we need to sort it into descending order. One way we can do this is with a function;

numberArray.sort(*function* (*a*, *b*) {

return *a* - *b*;

});

4. We now need to create a function that takes the highest value which will be at index value 0 and subtract it from the lowest value, which will be at -1 the array length. Remember that arrays are index 0. So if the array has 10 elements, the final position will be 9.

The first position is numberArray[0] and the final position is numberArray[numberArray.length-1]. We can now create a simple calculation that performs this task.

*let* amplitude = *function* () {

*let* amp = numberArray[0] - numberArray[numberArray.length - 1];

if (amp <= 0) {

return amp \* -1;

} else {

return amp;

}

};

Here we have created a function amplitude. Notice that there are no arguments between the brackets after function. This is because the function will take data from the global scope. We will not pass information into it.

We have declared the variable ‘amp’ within the function that performs our calculation.

Because we want the difference between the biggest number and smallest number; the amplitude should ALWAYS be positive. We can then use a simple if statement to multiply by -1 (to make our calculation positive) if amp is less than 1. Or to simply return our amplitude if it is a value greater than 1.

**Final Code Block**

// Problem:

//We work for a company building a smart home thermometer. Our most recent task is this: "Given an array of temperatures of one day, calculate the temperature amplitude. Keep in mind that sometimes there might be a sensor error."

*const* temperatures = [

3,

-2,

-6,

-1,

"error",

9,

1000,

-101,

13,

17,

15,

14,

9,

5,

];

*let* numberArray = [];

for (*let* i = 0; i < temperatures.length; i++) {

if (

typeof temperatures[i] === "number" &&

temperatures[i] <= 100 &&

temperatures[i] >= -100

) {

numberArray.push(temperatures[i]);

} else {

console.log(`These are the errors: ${temperatures[i]}`);

}

}

numberArray.sort(*function* (*a*, *b*) {

return *a* - *b*;

});

*let* amplitude = *function* () {

*let* amp = numberArray[0] - numberArray[numberArray.length - 1];

if (amp <= 0) {

return amp \* -1;

} else {

return amp;

}

};

console.log(`The temperature amplitude is; ${amplitude()} degrees.`);

**Alternative Solution**

*const* calcTempAmplitude = *function* (*temps*) {

*let* max = *temps*[0];

*let* min = *temps*[0];

for (*let* i = 0; i < *temps*.length; i++) {

*const* curTemp = *temps*[i];

if (typeof curTemp !== "number") continue;

if (curTemp > max) max = curTemp;

if (curTemp < min) min = curTemp;

}

console.log(max, min);

};

calcTempAmplitude([3, 7, 2, 3, 1, 6]);

1. We create a calctemptAmplitude function. We set it so that we can pass in ‘temps’ as an argument.
2. We set the max value to be at index position 0. This value will change.
3. We set the min value to be at index position 0. Remember that this value will also change.
4. Create a for loop that will iterate through the data in ‘temps’. Notice that this will iterate through data that is NOT in the global scope, only in the local scope.
5. Create a variable that we can then perform some logic on; ‘const curTemp = temps[i]’ which simply means that curTemp is the temperature at position ‘i’ as we iterate through the data set.
6. Create logic that ignores the value if it is not a number; ‘typeof curTemp !== ‘number’) continue; Simply put, if the information at position ‘I’ is not a number we will ‘continue’ without performing any further calculations.
7. If the curtemp is greater than our current max value, that will then become the new max value.
8. If the curtemp is less than our current min value, then that will become the new min value.