

1 hour

## Task 1: Kelvin Working Conditions

You have been employed by Kelvin, the mad scientist, to create an application that will alert him in simple, plain English, whether the weather is good enough for him to work on building his new lab somewhere within the boundary of his secret location.

He has sensors set up to detect the temperature (now in Celsius, thanks to you!) and wants to display a message on one of his terminals based on that temperature.

Follow the instructions below:

1. Create 3 variables, one named `temperature`, another named `condition`, and the last one named `goToWork` - all declared without a value.
2. Ask the user for the value of `temperature` and store that value within the variable of the same name.
3. Next, do the same but for the `condition` variable. Anticipated values for condition are:
  - a. Sunny
  - b. Cloudy
  - c. Raining
  - d. Snowing
  - e. Thunder
4. This next part can be done in a number of ways, either as nested-if-statements, or as a case-switch.
  - a. Should the temperature be **10 or above** and the condition is...
    - i. ... sunny – `goToWork` should be true
    - ii. ... cloudy – `goToWork` should be true
    - iii. ... raining – `goToWork` should be false as no-one likes being wet and cold
    - iv. ... snowing – `goToWork` should be false
    - v. ... thunder – `goToWork` should be false as Kelvin is working with metal
  - b. Should the temperature be **above 20** and the condition is...
    - i. ... sunny – `goToWork` should be true
    - ii. ... cloudy – `goToWork` should be true
    - iii. ... raining – `goToWork` should be true as being warm and wet isn't the worst
    - iv. ... snowing – `goToWork` should be false as the weather is too weird
    - v. ... thunder – `goToWork` should be false as Kelvin is working with metal
  - c. Should the temperature be **above 30** and the condition is...
    - i. ... sunny – `goToWork` should be false as it's too hot to work effectively
    - ii. ... cloudy – `goToWork` should be false as it's too hot to work effectively
    - iii. ... raining – `goToWork` should be false as it's too hot to work effectively
    - iv. ... snowing – `goToWork` should be false as the weather is too weird
    - v. ... thunder – `goToWork` should be false as it's too hot to work effectively
  - d. Should the temperature be **below 10** and the condition is...
    - i. ... sunny – `goToWork` should be false as it's too cold to work effectively
    - ii. ... cloudy – `goToWork` should be false as it's too cold to work effectively
    - iii. ... raining – `goToWork` should be false as it's too cold to work effectively
    - iv. ... snowing – `goToWork` should be false as it's too cold to work effectively
    - v. ... thunder – `goToWork` should be false as it's too cold to work effectively
5. Depending on the outcome of `goToWork`, display a message on the screen that tells Kelvin whether the weather is fit for purpose, and he can continue construction on his new lab.
6. If you tried the above as nested if-statements, try it again using the case-switch method instead. You may find it becomes easier to read and is easier to write.

**STRETCH CHALLENGE:** Move the evaluation of the weather conditions into a function and return the resulting value to a variable which is then evaluated to write a message to the console. Wrap the input and evaluation in a loop so that we can simulate continuous changing in weather conditions.

## WEEK 6: HLT (JAVASCRIPT SYNTAX PART 2)

|  |                                     |
|--|-------------------------------------|
| 1 hour   | <b>Task 2: Arrays and For Loops</b> |
| <p>There's a lot to learn in JavaScript and unfortunately, we won't get time to look at all during our weekly sessions.</p> <p>Please watch the following video on <a href="#">arrays</a>, the video on <a href="#">For Loops</a>, and the video on <a href="#">For-Of and For-Each Loops</a>.</p> <p><b>Code along with these videos and play around with the values so that you familiarize yourself with what is happening within the code.</b></p> <p>You may also wish to look at supporting material on W3School or YouTube. You'll need to understand how these are used before moving on to the next task! 😊</p> |                                     |

1 hour

## Task 3: Fruity Loops

You have been given a contract for the fruit shop again, and they are wanting to make a display with names of the fruit written above them. They can order in nice, pretty letters from a crafts supply depot but need to know how many letters to be ordering.

Follow the instructions below:

1. Create an array called `fruits` with the elements:
  - a. Apple
  - b. Orange
  - c. Banana
  - d. Pear
  - e. Peach
  - f. Strawberry
  - g. Cherry
  - h. Acai
2. Iterate through that array using a `for-in` loop using `let fruit in fruits`.
3. Within that loop, create 2 variables called `vowels` and `consonants`, and set their value to `0`.
4. Within the loop, create another loop, but this time use a `for` loop using `i` as the iterator.
5. This for loop will want to look through every letter of the fruit and determine how many vowels and how many consonants are within that word.
  - a. Do this by looking at the letter with `fruits[fruit][i]`
  - b. Then comparing the letter to either a `case-switch` or `nested-if`
  - c. You could even make a `method` that has the `case-switch` or `nested-if` and `return` the value!
  - d. If the value is 'A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o', or 'u', add 1 to the `vowel` variable.
  - e. If the value is anything else, add 1 to the `consonants` variable.
6. For each element within the array, print out to console:
  - a. "Apple has 2 vowels and one consonant" etc

Be careful with this task as you may need to consider how to handle the upper-case and lower-case values separately. You can force the examination of the letter to lowercase using `.toLowerCase()`

**STRETCH CHALLENGE:** Change your print statements to start with the correct syntax. A sentence that starts with a vowel should open a sentence with "An", and a sentence that starts with a consonant should open a sentence with "A":

An Apple has...

A Banana has...

Furthermore, should the fruit only have one vowel (such as cherry) or one consonant (such as acai), the message should utilise singular and plural properly too:

A Cherry has 1 vowel and 5 consonants

An Acai has 3 vowels and 1 consonant

**STRETCH CHALLENGE:** Tally up the total number of each letter so that the fruit shop can order the right letters from the supply depot. Be careful of capital letters too - this may change the way you are evaluating each letter!

1 hour

## Task 4: Calculator

You are tasked with creating re-usable methods that can be used throughout the business. The business needs you to create methods for mathematics operations.

Follow the instructions below:

1. Ask the user for a number with a `prompt()` and store the value in a variable called `firstValue`
2. Ask the user for a second number with a `prompt()` and store that value in a variable called `secondValue`
3. Ask the user for a third input with `prompt()` storing the value in a variable called `operation`. Expected operations are:
  - a. `+`  
This is the addition symbol, located next to the backspace key (hold shift)
  - b. `-`  
This is the subtraction symbol, located next to number 0 key (hold shift)
  - c. `/`  
This is the division symbol, a forward slash, located next to the full stop key
  - d. `*`  
This is the multiplication symbol, a star, accessed by holding shift and pressing number 8
  - e. `^`  
This is the to-the-power-of symbol, known as a caret in programming, accessed by holding shift and pressing the number 6
4. Write a method for the 5 operations listed above (one for each operation) that takes in both values and returns the result of the operation.
  - a. For example `function multiplication(firstValue, secondValue) { return firstValue * secondValue; }`
5. Create a case-switch for evaluating the operation the user supplied, and depending on the value, execute the relevant function.
6. Print out to console `firstValue`, `operator`, `secondValue`, an equal sign, and the answer:
  - a. `2 x 8 = 16`

**STRETCH CHALLENGE:** Wrap the code in a continuous loop that only ends when the user responds to a prompt that asks them “would you like to do another calculation?” with their answer being “no”.

**STRETCH CHALLENGE:** Change the above code to also include methods for processing `sin`, `cos`, and `tan`. You can use the methods `Math.sin(x)`, `Math.cos(x)`, `Math.tan(x)` but be aware that the user only needs to supply a single value and the operation they wish to do when needing `sin`, `cos`, and `tan`!

## WEEK 6: HLT (JAVASCRIPT SYNTAX PART 2)

### Marking Criteria Task 1/3/4

|                     | Pass  | Merit   | Distinction   |
|---------------------|---|---|---|
| <b>Syntax</b>       | <ul style="list-style-type: none"> <li>Attempts to use JS syntax with some success</li> </ul>   | <ul style="list-style-type: none"> <li>JS syntax is largely accurate with some errors</li> </ul>                                | <ul style="list-style-type: none"> <li>JS syntax is consistently accurate and appropriate to the task</li> </ul>                          |
| <b>Presentation</b> | <ul style="list-style-type: none"> <li>Some whitespace used to good effect</li> <li>Indentation attempted but inaccurate</li> <li>Correct HTML naming convention</li> </ul> | <ul style="list-style-type: none"> <li>Whitespace used appropriately, at times</li> <li>Indentation largely accurate</li> </ul> | <ul style="list-style-type: none"> <li>Clear commentary provided throughout</li> <li>Code is explicitly clean and easy to read</li> </ul> |

### Marking criteria Task 1

|             | Pass  | Merit   | Distinction   |
|-------------|---|---|---|
| <b>Code</b> | <ul style="list-style-type: none"> <li>Attempted writing out the project</li> <li>Declares the 3 variables</li> <li>Prints to the screen whether Kelvin should go to work or not</li> </ul> | <ul style="list-style-type: none"> <li>Either: <ul style="list-style-type: none"> <li>Uses a case-switch to analyse the temperature and the condition and gives a response depending on the two values but not to the project description</li> <li>Uses if-statements to analyse the temperature and the condition and gives the correct response depending on the two values according to the project description</li> </ul> </li> <li>Hard-codes in the values for temperature and condition</li> </ul> | <ul style="list-style-type: none"> <li>Uses case switching to analyse the two values of temperature and condition</li> <li>Accurately details the correct response according to the project brief</li> <li>Takes in values from the user input</li> </ul> |

### Marking criteria Task 3

|             | Pass   | Merit   | Distinction  |
|-------------|--|---|--|
| <b>Code</b> | <ul style="list-style-type: none"> <li>Stores a single fruit name in a variable (not an array)</li> <li>Attempts to count the number of vowels and consonants within the variable</li> </ul> | <ul style="list-style-type: none"> <li>Creates an array of fruits</li> <li>Uses any form of loop to iterate over the list of values within the array</li> <li>Counts the correct number of vowels and consonants within the fruit but in an inefficient way (if-statement followed by if-statement etc.)</li> </ul> | <ul style="list-style-type: none"> <li>Creates an array of fruits</li> <li>Uses 'for fruit in fruits' loop</li> <li>Creates variables for vowels and consonants</li> <li>Compares the letter of each fruit with vowels, either as if statements using the or operator within the expression (  ), or by using a case switch statement with all cases pointing to a single line that increases the vowels by 1. Default case to increase consonants by 1</li> <li>Prints the correct message to the console</li> <li>Handles capital letters</li> </ul> |

## WEEK 6: HLT (JAVASCRIPT SYNTAX PART 2)

### Marking criteria Task 4

|      | Pass  | Merit   | Distinction  |
|------|---|---|--|
| Code | <ul style="list-style-type: none"> <li>• Hard codes in firstValue, secondValue, and operator</li> <li>• Uses if statements to analyse the operator and provides an answer to console</li> <li>• Uses the basic 4 operators, (+, -, /, *)</li> </ul> | <ul style="list-style-type: none"> <li>• Accepts prompts for firstValue, secondValue, and an operation</li> <li>• Does not handle errors when the user inputs invalid values (input a word instead of a number, input something that isn't one of the 5 symbols for operators)</li> <li>• Uses if statements to identify the symbol being used</li> <li>• Points to methods that print the answer to console</li> <li>• Prints the correct value to the screen in any format</li> </ul> | <ul style="list-style-type: none"> <li>• Accepts prompts for firstValue, secondValue, and an operation</li> <li>• Handles errors when the user inputs invalid values (input a word instead of a number, input something that isn't one of the 5 symbols for operators)</li> <li>• Uses a case switch to identify which of the symbols is being used, and pointing to the correct method</li> <li>• Correctly builds a method for each operator</li> <li>• Returns a value to the main program from each method and having only one print to console statement (not having a print to console statement within each method)</li> <li>• Prints the correct value for the sum being requested of the program using template literals (back tick and variable supplementation `hello \${name}`)</li> </ul> |