

Session 10

Consolidation Day

Katas



Challenge: Objects Equal?

Write a program to compare two objects to determine if the first one contains the same properties as the other (which may contain additional properties).

For example, given

```
const firstObj = { a: 1, b: 2, c: 3 };  
const secondObj = { a: 1, b: 2, c: 3 };  
const thirdObj = { a: 1, b: 2, d: 3 };
```

`firstObj` and `secondObj` will return true as they both contain the same properties where as `firstObj` and `thirdObj` will return false, as there is no `d` property in `firstObj`.



Challenge: CSV

Write a program to convert comma-separated values (CSV), into a 2D array. A new line indicates a new row in the array

Example input:

```
abc,def,ghi  
jkl,mno,pqr  
stu,vwx,yza
```



Challenge: Seven Boom

Create a function that takes an array of numbers and returns the word 'Boom!' if the number 7 appears in the array. Otherwise return 'No booming here'.

Examples:

```
sevenBoom([1, 2, 3, 5, 7]) => 'Boom!'
sevenBoom([1, 2, 3, 5, 9]) => 'No booming here'
sevenBoom([1, 2, 3, 5, 97]) => 'Boom!'
```



Challenge: ISBN 13

Given an ISBN 13 digit, calculate if it is a genuine code or not. Assume you can receive ISBNs with space and hyphens (or without). The last digit is the check digit.

To calculate the check digit, simply multiply the first 12 digits by 1 and 3, left to right.

Once you have sum, use a modulo of 10. If the remainder is zero, the check digit is zero, if not subtract that number by 10. This will then be the check digit.



Challenge: ISBN 10

Given an ISBN 10 digit, calculate if it is a genuine code or not. Assume you can receive ISBNs with space and hyphens (or without). The last digit is the check digit.

To calculate the check digit, simply follow these steps:

1. Extract the first 9 digits.
2. Working left to right, multiply the first by 9, the second by 8 etc.
3. Obtain the sum of the values accumulated from step 2.
4. From this sum, use a modulo of 11 and obtain the remainder.
5. Subtract 11 from this remainder to then produce the check digit.

If the check digit is 10, then this is converted to an 'X'.



Challenge: On Par

Given a set of defined golf course holes and their respective shot counts, work out the score for a player when they complete 18 rounds of golf.

For example, an 18 round course has the following scoring holes per round:

```
4,4,5 ...
```

A player's score can fluctuate between 2 shots to 5, so for the user in this challenge their 18 hole shots per round were:

```
4, 5, 3, 4 ...
```

Extension: Also display the shot played as the judges don't believe the score to be genuine

Extension: Your score calculator needs to be used on different golf courses which may have different scores per hole. Update your solution to make the logic re-usable for any type of golf course and any user scores.



Challenge: Sum of Odd Numbers

Given the following pyramid of odd numbers:

```

      1
    3 5
  7 9 11
13 15 17 19
21 23 25 27 29
```

Write a function that returns the sum of the given row number, i.e: row 1 will be 1, row 2 8 (3 + 5) etc.



Challenge: Simpsons

Given an array containing hashes of names, return a string formatted as a list of names separated by commas except for the last two names which should be separated by an ampersand.

Example:

```
List: [ { name: 'Bart' }, { name: 'Lisa' }, { name: 'Maggie' } ]
// returns 'Bart, Lisa & Maggie'
```

```
List: [ { name: 'Bart' }, { name: 'Lisa' }]  
// returns 'Bart & Lisa'  
  
List: [ { name: 'Bart' }]  
// returns 'Bart'
```



Challenge: Divisor

Write a function that takes an integer **n** where **n > 1** and returns an array with all the integer's divisors (except for **1** and the number itself), from smallest to largest. If the number is prime, return the string 'X is prime'

Example:

```
divisors(12); // returns [2, 3, 4, 6]  
divisors(25); // returns [5]  
divisors(13); // returns '13 is prime'
```



Challenge: Spin Words

Write a function that takes a string of words and reverses all words with a length of 5 or more.

Example:

```
spinWords('Hey fellow coders'); // returns 'Hey wollef sredoc'
```



Challenge: Baking

Given you are provided some recipes and ingredients to bake some cakes, work out how many cakes in fact can be baked.

Your function should take **recipe** (an object) and **ingredient** (an object) and should return the maximum number of cakes that can be baked as an integer. For simplicity, there are no units of food we are going to deal with (e.g. 1lb of flour or 200g of sugar are simply **1** and **200**). Ingredients that are not present in the objects can be considered as **0**.

Example:

```
cakes({flour: 500, sugar: 200, eggs: 1}, {flour: 1200, sugar: 1200, eggs: 5})  
// returns 2  
  
cakes({apples: 3, flour: 300, sugar: 150, milk: 100, oil: 100}, {flour: 2000, sugar:  
500, milk: 2000})  
// returns 0
```