**A Set in Mathematics**

If you think back to mathematics, a set is a collection of distinct items. For example, {2, 4, 5, 6} is a set because each number is unique and appears only once. However, {1, 1, 2, 4} is *not* a set because it *contains duplicate entries* (the 1 is in there more than once!).

In JavaScript, we can already represent something similar to a mathematical set using an array.

**const** nums = [2, 4, 5, 6];

However, arrays *do not enforce items to be unique*. If we try to add another 2 to nums, JavaScript won't complain and will add it without any issue.

nums.push(2);

console.log(nums);

*[2, 4, 5, 6, 2]*

…and now nums is no longer a set in the mathematical sense.

**Sets**

In ES6, there’s a new built-in object that behaves like a mathematical set and works similarly to an array. This new object is conveniently called a "Set". The biggest differences between a set and an array are:

* Sets are not indexed-based - you do not refer to items in a set based on their position in the set
* items in a Set can’t be accessed individually

Basically, a Set is an object that lets you store unique items. You can add items to a Set, remove items from a Set, and loop over a Set. These items can be either primitive values or objects.

**How to Create a Set**

There’s a couple of different ways to create a Set. The first way, is pretty straightforward:

**const** games = **new** Set();

console.log(games);

*Set {}*

This creates an empty Set games with no items.

If you want to create a Set from a list of values, you use an array:

**const** games = **new** Set(['Super Mario Bros.', 'Banjo-Kazooie', 'Mario Kart', 'Super Mario Bros.']);

console.log(games);

*Set {'Super Mario Bros.', 'Banjo-Kazooie', 'Mario Kart'}*

Notice the example above automatically removes the duplicate entry "Super Mario Bros." when the Set is created. Pretty neat!

**QUIZ QUESTION**

Select the collections below that represent a Set in JavaScript.

* {1, 'Basketball', true, false, '1'}
* {}
* 

{1, 1, 1, 1}

* 

{false, '0', 0, 'Soccer', 3.14, 25, 0}

* {'Gymnastics', 'Swimming', 2}