Menu

How to Compare 2 Objects in JavaScript 

Objects are reference types so you can't just use `===` or `==` to compare 2 objects. One quick way to compare if 2 objects have the same key value, is using `JSON.stringify` . Another way is using Lodash `isEqual` function 👏

```
const k1 = {fruit: '🥝'};
const k2 = {fruit: '🥝'};

// Using JavaScript
JSON.stringify(k1) === JSON.stringify(k2); // true

// Using Lodash
_.isEqual(k1, k2); // true
```

Deep Nested Comparison

Yup, the 2 ways also work for deep nested objects.

```
const one = {
  fruit: '🥝',
  nutrients: {
    energy: '255kJ',
    minerals: {
      name: 'calcium'
    }
```

```
  }
};

const two = {
  fruit: '🥝',
  nutrients: {
    energy: '255kJ',
    minerals: {
      name: 'calcium'
    }
  }
};

// Using JavaScript
JSON.stringify(one) === JSON.stringify(two); // true

// Using Lodash
_.isEqual(one, two); // true
```

Which one should I use?

Well that depends. For `JSON.stringify()`, the order matters. So if the key-value pair are ordered differently in the two objects but are the same, it will return false. Whereas it doesn't matter in Lodash `isEqual`, it will return true as along as the key-value pair exists.

Here's my recommendation. For a quick and dirty solution, I'd use `JSON.stringify()`. But for a more robust solution that cover more of those odd edge cases, use the Lodash way.

```
const one = {
  fruit: '🥝',
  energy: '255kJ',
```

```
};

const two = {
  energy: '255kJ',
  fruit: '🥝',
};


// Using JavaScript
JSON.stringify(one) === JSON.stringify(two); // false

// Using Lodash
_.isEqual(one, two); // true
```

Community Suggestions

ES6 Way for comparing 2 objects

This is a solution suggested by @mustafauzun0. Few things to note though, it won't work with nested objects and the order of the keys are important. The idea behind this is similar to the stringify way. It coverts the object into a string and compare if the strings are a match. Essentially it's comparing the equality of two strings. That's why the order matters.

```
const k1 = {fruit: '🥝'};
const k2 = {fruit: '🥝'};

Object.entries(k1).toString() === Object.entries(k2).toString();
// true
```

Thanks: @mustafauzun0

JSON.stringify vs Lodash's **isEqual** Performance

Cole Turner: Worth noting that objects don't guarantee sort order, and stringify is going to cost more in performance because it has to serialize the whole object whereas lodash can exit early if it finds a mismatched key. This comes up in interviews ☺

Thanks: @coleturner

blnkdotspace: Lodash is considerably faster. Because lodash will quit soon as it reaches the first difference, but stringify goes right till the end unnecessarily. If you have small arrays it's okay but for my personal use cases with more than 1000 events/products in a list I wouldn't ever use Stringify.

Thanks: @blnkdotspace

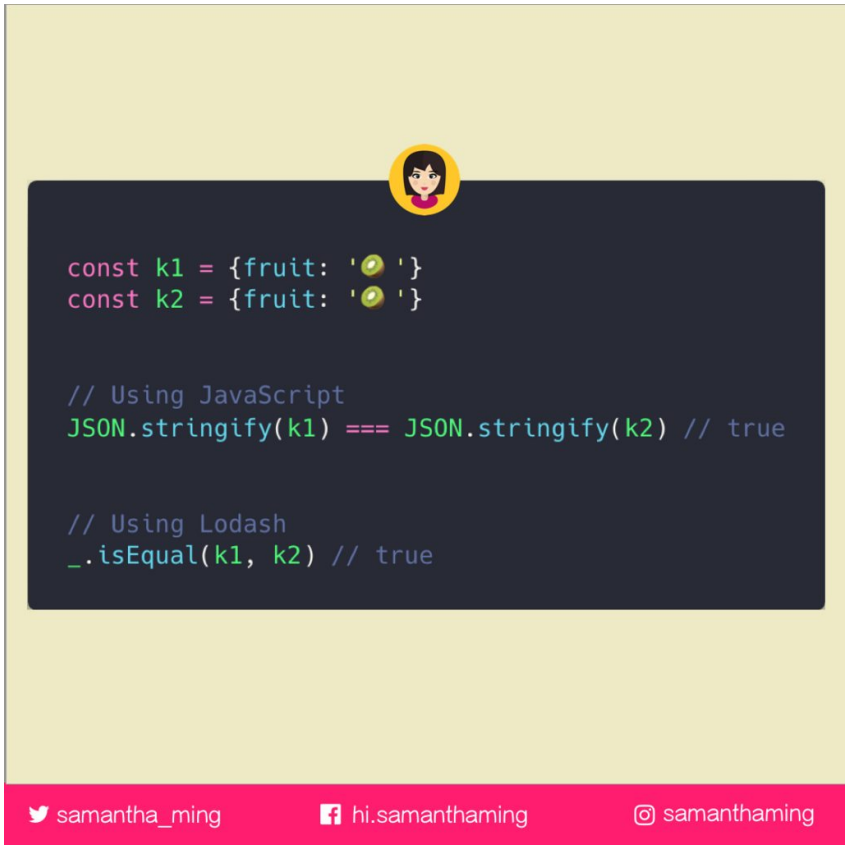
Resources

- MDN Web Docs - JSON.stringify
- Lodash: isEqual
- Understanding JavaScript Objects
- Object Equality in JavaScript
- JavaScript deep object comparison - JSON.stringify vs deepEqual
- Stack overflow: Is it fine to use JSON.stringify for deep comparisons and cloning?
- How to determine equality for two JavaScript objects?

Like ♡

Like this on Twitter

Like this on Instagram



What I use

If you're interested in what I use to create this code tidbit, I list them here.

‹ More Code Tidbits

HOME
ABOUT
CODE TIDBITS
BLOG
PODCAST
RECOMMENDATIONS
CONTACT

What I use
Courses I Love
Podcasts I Love
Tech Stack
Invite Me to Speak